МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»
Кафедра систем информационной безопасности

«ДОПУЩЕН К ЗАЩИТЕ»
Зав. кафедрой к.п.н., доцент Р. Ш. Бердибаев
_____ «_____» _____2019 г.
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Исследование и обратная разработка кода с целью обнаружения уязвимостей
Специальность: 5В100200 – «Системы информационной безопасности»
Выполнила: Медведева Екатерина Евгеньевна Группа СИБ-15-3
Научный руководитель: Сатимова Елена Григорьевна

Консультанты:
по экономической части:
к.э.н., профессор Арисбаева М.Т.
(ученая степень, звание, Ф.И.О)
подпись «28» мая 2019 г.
(подпись)

по безопасности жизнедеятельности:
д.т.н., стар. преп. Бекбаганов Ш.Ш.
(ученая степень, звание, Ф.И.О)
_____ «28» мая 2019 г.
(подпись)

по применению вычислительной техники:
к.т.н., доцент Сатимова Е.Г.
(ученая степень, звание, Ф.И.О)
_____ «5» июня 2019 г.
(подпись)

Нормоконтролер:
Ст. преподаватель Аскарова Ф.Б.
(ученая степень, звание, Ф.И.О)
_____ «5» июня 2019 г.
(подпись)

Рецензент:
_____
(ученая степень, звание, Ф.И.О)
_____ «_____» _____2019 г.
(подпись)

Алматы 2019

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»

Институт систем управления и информационных технологий

Кафедра систем информационной безопасности

Специальность 5В100200 – «Системы информационной безопасности»

ЗАДАНИЕ
на выполнение дипломного проекта

Студенту Медведевой Екатерине Евгеньевне

Тема проекта Исследование и обратная разработка кода с целью обнаружения уязвимостей

Утверждена приказом по университету №_124_ от «_26_» _10_ 2018 г.

Срок сдачи законченного проекта «_6_» _июня_ 2019 г.

Исходные данные к проекту (требуемые параметры результатов исследования (проектирования) и исходные данные объекта): проекте была произведена обратная разработка вредоносного кода, а именно вируса – шифровальщика. На основании проделанной работы, была выявлена степень ущерба, нанесенная данным вредоносным программным обеспечением. Было определено какие алгоритмы шифрования используются в данном средстве, и рассмотрены все внутренние процессы, происходящие при запуске вируса - шифровальщика.

Перечень вопросов, подлежащих разработке в дипломном проекте, или краткое содержание дипломного проекта: дипломный проект включает в себя 5 глав, разделенных на подглавы, каждая из которых освещает определенную тематику, используемую при обратной разработке вируса - шифровальщика.

В первой главе дипломного проекта представлена общая информация по вирусам шифровальщика: типы, способы распространения, разновидности.

Во второй главе дипломного проекта представлены алгоритмы шифрования.

В третей главе описывается сам процесс обратной разработки кода

В четвертой главе приводится технико-экономическое обоснование, показывающее актуальность исследования вируса – шифровальщика о с финансовой точки зрения.

В пятой главе рассматриваются необходимые условия для комфортного исследования вируса- шифровальщика.

Основная рекомендуемая литература:

1Описание вируса – шифровальщика URL: https://www.anti-malware.ru/threats/virus-encoder

2 Аманжолова К. Б., Алибаева С. А. Экономика предприятии телекоммуникации: Учебное пособие. - Алматы: АИЭС, 2003

3 Голубицкая Е. А., Жигульская Г. М. Экономика связи. – М.: Радио и связь, 2000г.

4Методы заражения URL: http://comp-neo.ru/virus-shifrovalshchik-rasshifrovat-fajly

5Разновидности вирусов-шифровальщиков URL: https://habr.com/ru/post/206830/

6 Нормы микроклимата URL: http://adilet.zan.kz/rus/docs /V050003789

Консультанты по проекту с указанием относящихся к ним разделов проекта

Раздел	Консультант	Сроки	Подпись
Технико-экономическая обоснование	Алибаева Ж.С.	04.03-28.05	
Безопасность жизнедеятельности	Бекбосаров М.М.	04.03-28.05	
Вычислительной техники	Сатимова Е.Г	09.01-28.05	

График
подготовки дипломного проекта

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание
Вопрос, общее описание	1.02.2019 – 22.02.2019	
Программирование и документирование	23.02.19 – 20.03.2019	
Практическая разработка кода	21.03.19 – 24.04.2019	
Технико-экономическое обоснование	25.04.2019 – 28.05.2019	
Безопасность жизнедеятельности	10.04.2019 – 28.05.2019	
Выводы борьба с вирусами-шифровальщиками	29.05.2019 – 6.06.2019	

Дата выдачи задания «10» октября 2018 г.

Заведующий кафедрой _____ (_____)
(Подпись) (Ф.И.О)

Научный руководитель проекта _____ (Сатилова Б.Т.)
(Подпись) (Ф.И.О)

Задание принял к исполнению студент _____ (Мудуева Б.Е.)
(Подпись) (Ф.И.О)

АННОТАЦИЯ

В данном дипломном проекте была произведена обратная разработка вредоносного кода, а именно вируса – шифровальщика. На основании проделанной работы, была выявлена степень ущерба, нанесенная данным вредоносным программным обеспечением. Было определено какие алгоритмы шифрования используются в данном средстве, и рассмотрены все внутренние процессы, происходящие при запуске вируса - шифровальщика.

АҢДАТПА

Бұл тезис жобасында зиянды кодтың кері дамуы, атап айтқанда, вирус - криптограф. Жүргізілген жұмыстардың негізінде бұл зиянды бағдарлама келтірген зиянның мөлшері анықталды. Бұл құралда қандай шифрлеу алгоритмдері қолданылатынын және вирус іске қосылған кездегі барлық ішкі процестерді - шифрлау кодын қарастырған.

ANNOTATION

In this thesis project was carried out the reverse development of a malicious code, namely a virus - a cryptographer. Based on the work done, the extent of the damage caused by this malware was identified. It was determined what encryption algorithms are used in this tool, and all internal processes occurring when the virus is launched — the encryption code — are considered.

Содержание

Введение

Вирус-шифровальщик – это тип вредоносного ПО из криптовирологии, который угрожает опубликовать данные жертвы или навсегда заблокировать доступ к ним, если выкуп не будет выплачен. Криптовирология представляет собой науку, которая говорит о том, как использовать криптоанализ вредоносного программного обеспечения. Более продвинутые вредоносные программы используют технику, называемую криптовирусным вымогательством, в которой она шифрует файлы жертвы, делая их недоступными, и требует выкуп за дешифрование. В правильно реализованной атаке с использованием криптовирусного вымогательства восстановление файлов без ключа дешифрования является неразрешимой проблемой.

Атаки на пользователей, как правило, выполняются с использованием трояна, замаскированного под легитимный файл, который пользователь обманом загружает или открывает, когда он приходит в виде вложения электронной почты.

Каждый вирус-шифровальщик имеет свой алгоритм шифрования, а бывает даже несколько. Недостаточно знать лишь алгоритм, чтобы подобрать ключ дешифрования. Обнаружив исходный код вируса, можно лишь выяснить насколько глубоко было проникновение, и какие файлы системного уровня были подвержены инфицированию, какие данные были удалены, а какие изменены.

Обратная разработка кода способствует выяснению уровня ущерба, возможности восстановления, сложности шифрования и исследованию задетых процессы системы.

Главным и самым простым путем распространения вредоносного ПО на компьютеры пользователей является почта. Не подготовленные работники организаций или, ни о чем не подозревающие, простые пользователи открывают письма со «скрытыми ссылками», вложенными архивами, хранящими в себе вирус - шифровальщик, с гиперссылками на вредоносное ПО, скачивая и тем самым заражая свою систему.

1 Вирус, общее описание

Компьютерный вирус – это разновидность вредоносного программного обеспечения, которое при запуске копирует себя, изменяя другие компьютерные программы и вставляя свой собственный код. Когда эта репликация завершается успешно, считается, что зараженные области «заражены» компьютерным вирусом [1].

Авторы вирусов используют социальную инженерию и подробные знания об уязвимостях безопасности для первоначального заражения систем и распространения вируса. Подавляющее большинство вирусов предназначено для систем, работающих под управлением Microsoft Windows, использующих различные механизмы для заражения новых хостов и часто использующих сложные стратегии анти-обнаружения, скрытности для обхода антивирусного программного обеспечения.

Мотивы создания вирусов могут включать в себя поиск прибыли, желание отправить политическое сообщение, личное развлечение, демонстрация наличия уязвимости в программном обеспечении, саботаж и отказ в обслуживании или просто потому, что они хотят исследовать проблемы кибербезопасности, искусственная жизнь и эволюционные алгоритмы.

Вирус-шифровальщик представляет собой разновидность вредоносных программ-вымогателей, которые посредством шифрования файлов, данных и других различных видов информации пользователя в форме вымогания требует внесения определенной суммы за средство расшифровки [1].

Можно вынести три основных подхода шифровальщиков:

1) шифрование файлов;

2) шифрование и блокировка системы;

3) шифрование и помеха работе в браузерах.

Компьютерные вирусы в настоящее время наносят экономический ущерб в миллиарды долларов каждый год из-за сбоя системы, потери ресурсов компьютера, повреждения данных, увеличения затрат на обслуживание и т.д. В ответ были разработаны бесплатные антивирусные инструменты с открытым исходным кодом и возникла индустрия антивирусных программ, продающая или свободно распространяющая защиту от вирусов среди пользователей различных операционных систем.

1.1 Способы и пути распространения

Главный и самый простой путь распространения вредоносного ПО на компьютеры пользователей является почта. Не подготовленные работники организаций или, ни о чем не подозревающие, простые пользователи открывают письма со «скрытыми ссылками», вложенными архивами, хранящими в себе вирус-шифровальщик, с гиперссылками на вредоносное ПО, скачивая и тем самым заражая свою систему [1].

Часто злоумышленники подстраивают содержание письма под того, кому оно направлено. Если это организация, то содержимое включает в себя вопросы относительно рабочих моментов, если это обычный пользователь, то чаще всего это реклама, предложение новой версии ПО, официальное приложение банка. Были зафиксированы случаи, когда вирус устанавливался под видом обновления на всем известный Adobe Flash Player, или использовались другие аспекты социальной инженерии.

Социальная инженерия играет не маловажную роль в распространении вирусов, ведь крайне важно для злоумышленника вызвать доверие и желание жертвы «кликнуть» по нужной ссылке.

Помимо почтовых сервисов, для распространения шифровальщиков активно используют:

1) социальные сети (спам-рассылка со взломанных аккаунтов друзей, знакомых);

2) зараженные веб-ресурсы;

3) баннеры (реклама);

4) сайты-распространители ломанных ПО, кейгенов (сайты-варезники);

5) сайты для взрослых;

6) магазины предложений и контента.

Также проводниками вирусов шифровальщиков служат и другие вредоносные программы. К примеру, трояны-бэкдоры способствуют получению злоумышленником удаленного доступа к системе посредством использования уязвимостей. Запуск шифровальщика зависит от того, когда злоумышленник проникнет в зараженное устройство. В таких случаях сложно выяснить откуда пришло «заражение», так как время заражения не совпадает с потенциально опасными действиями пользователя.

Вирус-шифровальщик, который был исследован в данном дипломном проекте, содержался в архиве и передавался по почте (фишинговой рассылкой). Пользователь, скачивая вложение в виде вируса, становился лишь потенциальной жертвой, но из-за малой осведомленности пользователей в сфере безопасности, как правило, большинство разархивировали скачанное вложение, открывали .exe файл и тем самым становился жертвой, запустив работу шифровальщика.

1.2 Характеристика и вид вируса-шифровальщика

Семейство Ransomware можно поделить на два типа. Первый тип – это вирусы-блокировщики – программы, блокирующие пользователю доступ к ОС или браузеру. Для того чтобы разблокировать требуют умеренный выкуп, можно оплатить, к примеру, через SMS-сообщение или переведя денежные средства на электронный кошелек. Этот вид вымогателей обрел широкую популярность у киберпреступников, поскольку оказался весь прибыльным. Но у

популярности есть и плохая сторона, данная тема в связи со своим массовым распространением привлекла внимание антивирусных компаний и правоохранительных органов. В конечном счете киберпреступники были окружены с двух сторон. Передача выкупа, которая производилась через платежные системы стала сложно осуществимой, поскольку были изменены правила регулирования платежей. Тем самым государственные органы смогли усложнить злоумышленникам извлечение денежных средств и сделали эту деятельность более рискованной, некоторые преступники были посажены в тюрьму [2].

Антивирусные компании, заметив масштабность проблемы борьбы с блокировщиками, создали утилиты доступные всем пользователям, к примеру Kaspersky WindowsUnlocker. Это не искоренило проблему полностью, но помогло сдержать натиск со стороны злоумышленников и число заражений резко стало падать.

Второй тип – вирусы-шифровальщики, которые делятся на два вида:

- troldesh;
- shade.

В период распространения вирусов-шифровальщиков набирала популярность криптовалюта, а именно биткойны – платежная система, в которой практически невозможно отследить движение транзакций, а тем более их регулировать. Злоумышленниками был разработан новый план и придуман новый подход, в котором вместо ограничения доступа к браузеру или ОС, преступники начали производить шифрование файлов.

Личные данные имеют ценность в отличии от программ, их нельзя заменит, переустановив ОС. Поскольку злоумышленники используют стойшикое шифрование, то восстановление и дешифрация становятся достаточно трудоемкой работой, а нередко и невозможной. Что дает преступникам требовать неограниченный по ценовой категории выкуп.

Рассматриваемый в данном проекте вирус-шифровальщик относится к семейству ransomware вида troldesh. Столкнувшись с данным видом инфекций не сложно понять что в скором времени обнаружится либо зашифрованные файлы, включая документы, фотографии, видео, либо в наихудшем раскладе зашифрованная система. В случае, представленном в этом проекте вирусом шифровальщиком, были повреждены и системные файлы.

Troldesh, как и все вирусы, приходит без уведомления о своих планах, и как только пользователь дал разрешение на скачивание прикрепленного документа, вирус уже имеет возможность инфицировать систему. Незаметно для пользователя вирус проникает в рабочую станцию (компьютер) и создает папку. В этой папке он сохраняет ключ, который активируется и запустит распространение инфекции, как только пользователь кликнет по файлу чтоб его открыть. Программа-вымогатель предназначается для файлов персонального компьютера и применяет алгоритмы шифрования, такие как MDA5, SHA256,

которые делают файлы недоступными. Единственный способ получить к ним доступ, выплатить выкуп субъекту угрозы, следуя инструкциям, которые отображаются в зашифрованных файлах. Именно поэтому их называют вымогателями, потому что требуется оплата, для того чтобы решить проблему. Требуемый платеж также может быть потребован в криптовалюте, в большинстве случаев в биткойнах. Наиболее опасный тип вымогателей дает пользователям крайний срок для завершения платежа, в противном случае файлы могут быть потеряны навсегда. Когда файл зашифрован, его можно восстановить только с помощью ключа дешифрования или мощного компьютера. Последнее на самом деле недоступно для большинства пользователей, поэтому такие атаки становятся очень серьезной угрозой. Также вымогатель попытается заразить другие компьютеры в сети, к которой подключен зараженный хост, поэтому он также обладает свойствами червя. Ransomware классифицируется как тип киберпреступности, который иногда упоминается в разделе «Преступление как услуга», когда используется для вымогательства денег.

«Преступление как услуга» – это когда профессиональный преступник или группа преступников разрабатывают передовые инструменты, «наборы» и другие комплексные услуги, а затем отправляются для продажи или сдачи в аренду другим преступникам, которые обычно менее опытны. Это оказывает сильное влияние на мир преступности – и киберпреступность в частности – потому что это снижает планку неопытным злоумышленникам для реализации сложных кибератак и мошенничества.

Вирус Troldesh – это вирус типа вымогателей, который нацелен на компьютеры посредством спама в электронной почте. Этот вирус был признан одной из самых разрушительных и разрушительных киберугроз, известных с конца 2014 года. Цель этого типа злоумышленников - зашифровать файлы пользователей и потребовать оплату. Когда данные блокируются, к зашифрованным файлам добавляются расширения .7h9r, xtbl, .ytbl и .da_vinci_code, no_more_ransom, .better_call_saul, .heisenberg и .windows10 [2].

Вирус блокирует изображения, видеоматериалы, документы, музыкальные файлы и другие данные, которые он находит на компьютере, с использованием метода шифрования SHA256, MDA5.

Troldesh использует два метода для получения выкупа: через адрес электронной почты и сервер Tor. В последнем случае пользователи должны получить доступ к анонимному браузеру, а затем следовать инструкциям для перевода денег. Даже если данные имеют жизненно важное значение, не платите деньги, поскольку это не гарантирует, что служба расшифровки вредоносного ПО Troldesh поможет вам восстановить заблокированную информацию. На рисунке 1 предоставлена статистика распространенности вирусов-вымогателей, где можно увидеть, что 22% приходится на вирус типа Troldesh [2].

Преступники следят за пользователем компьютера, чтобы решить, сколько денег у него попросить в обмен на ключ расшифровки; кроме того, они изменяют примечание о выкупе для отдельных жертв, как только выясняют, какие угрозы использовать против конкретных пользователей компьютеров.

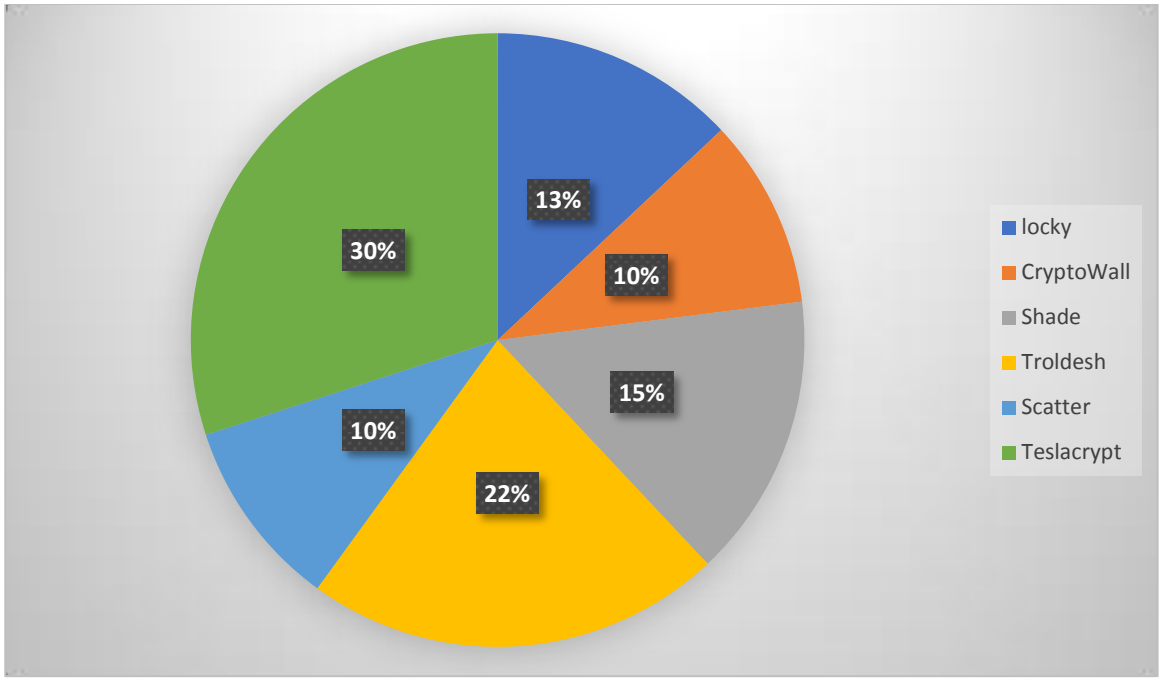

Рисунок 1 – Статистика распространения программ-вымогателей

Программы – вымогатели, относящиеся к семейству ransomware, выглядят как простой компьютерный вирус или сетевой червь и распространение происходит посредством массовой рассылки, на сайтах, через вложение.

2 Шифрование и дешифрование

В криптографии шифрование – это процесс преобразования исходного сообщения или информации таким образом, что только авторизованные стороны могут получить к нему доступ, а те, кто не авторизован, не могут. Шифрование само по себе не предотвращает помехи, но запрещает понятное содержание потенциальному перехватчику. В схеме шифрования предполагаемая информация или сообщение, называемое открытым текстом, зашифровывается с использованием алгоритма шифрования – зашифрованного генерирующего зашифрованного текста, который может быть прочитан только при расшифровке. По техническим условиям в схеме шифрования для генерации ключей о используются псевдослучайные числа. Авторизованный получатель может легко расшифровать сообщение с помощью ключа, предоставленного отправителем получателям [3].

На сегодняшний день надежность шифрования обычно измеряется размером ключа. Независимо от того, насколько силен алгоритм, зашифрованные данные могут подвергаться атакам методом "грубой силы", в которых пробуются все возможные комбинации ключей. В конце концов шифрование может быть взломано. Для большинства современных шифров с приличной длиной ключа время взлома их грубой силой измеряется тысячелетиями. Однако нераскрытый недостаток в алгоритме или прогресс в компьютерных технологиях или математических методах может резко сократить эти времена.

Обычно считается, что длина ключа должна подходить для обеспечения безопасности данных в течение разумного периода времени. Если предмет очень актуален, например, сообщения на полях сражений или ежедневная информация о запасах, то шифр, который защищает его в течение нескольких недель или месяцев вполне подойдет. Тем не менее, что-то вроде номера кредитной карты или секретов национальной безопасности должно храниться в течение более длительного периода, фактически навсегда. Таким образом, использование более слабых алгоритмов шифрования или более коротких ключей для некоторых вещей вполне допустимо, если срок полезности информации для постороннего лица истекает в короткий промежуток времени.

В процессе разбора кода вируса-шифровальщика, рассматриваемом в данном проекте, было определено, что используется два алгоритма шифрования - MDA5, SHA256. Каждый из этих алгоритмов имеет свои свойства, и свои сложности и именно наложение двух алгоритмов делается вирус фактически не дешифруемым.

Модель взаимодействия и передачи ключей между злоумышленником и жертвой представлены на рисунке 2.

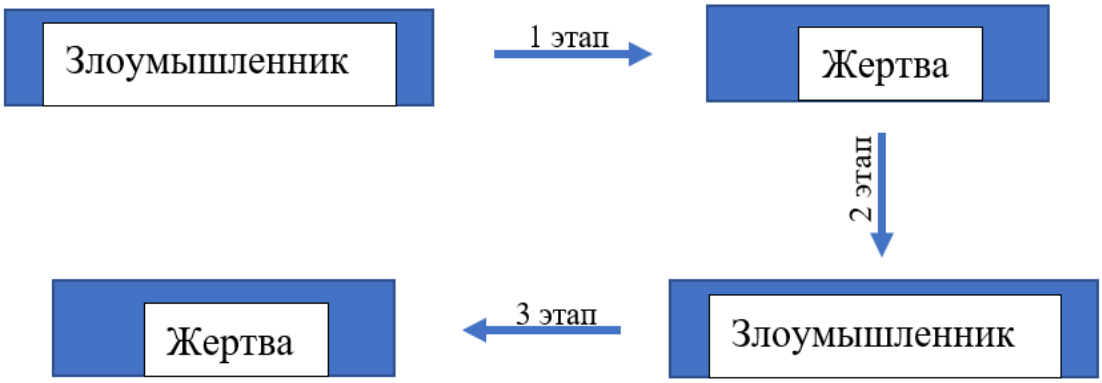

Рисунок 2 – Алгоритм взаимодействия жертвы и злоумышленника

Алгоритм взаимодействия жертвы и злоумышленника состоит из трех этапов:

1 этап – Злоумышленник генерирует пару ключей и помещает соответствующий открытый ключ в вредоносную программу. Вредонос выпущен.

2 этап – Для осуществления криптовирусной атаки вымогательства вредоносная программа генерирует случайный симметричный ключ и шифрует с его помощью данные жертвы. Он использует открытый ключ в вредоносной программе для шифрования симметричного ключа. Это называется гибридным шифрованием и приводит к небольшому асимметричному зашифрованному тексту, а также к симметричному зашифрованному тексту данных жертвы. Он обнуляет симметричный ключ и исходные текстовые данные, чтобы предотвратить восстановление. Он выдает пользователю сообщение, содержащее зашифрованный текст и информацию о том, как заплатить выкуп. Жертва отправляет злоумышленнику зашифрованный текст и электронные платежи.

3 этап – Злоумышленник получает платеж, расшифровывает зашифрованный текст с помощью личного ключа злоумышленника и отправляет ключ жертве. Жертва дешифрует зашифрованные данные с помощью необходимого симметричного ключа, тем самым завершив криптовирусную атаку [4].

Для шифрования вирус ищет файлы, хранящиеся на фиксированных, съемных и удаленных дисках. Затем он шифрует файлы со следующими расширениями и переименовывает файлы в [ИМЯ ФАЙЛА].Xtbl:

- .1cd;
- .3ds;
- .3fr;
- .3g2;
- .3gp;
- .7z;
- .accda;
- .accdb;

- .accdc;
- .accde;
- .accdt;
- .accdw;
- .adb;
- .adp;
- .ai;
- .ai3;
- .ai4;
- .ai5;
- .ai6;
- .ai7;
- .ai8.

2.1 Алгоритм шифрования MDA5

MDA5 – это широко используемая хеш-функция, создающая 128-битное хеш-значение. Хотя MDA5 изначально был разработан для использования в качестве криптографической хеш-функции, было обнаружено, что он имеет множество уязвимостей. Но его все еще можно использовать в качестве контрольной суммы для проверки целостности данных, но только от непреднамеренного повреждения [5].

Алгоритм хеширования MDA5 является односторонней криптографической функцией, которая принимает сообщение любой длины в качестве входных данных и возвращает в качестве выходных данных дайджест-значение фиксированной длины, которое будет использоваться для аутентификации исходного сообщения.

Хэш-функция MDA5 изначально была разработана для использования в качестве безопасного криптографического алгоритма хеширования для аутентификации цифровых подписей. MDA5 устарела для использования, отличного от не криптографической контрольной суммы, для проверки целостности данных и обнаружения непреднамеренного повреждения данных.

Первоначально разработанный как алгоритм шифрования кода аутентификации сообщений для использования в Интернете, хеширование MD5 больше не считается надежным для использования в качестве криптографической контрольной суммы, поскольку исследователи продемонстрировали методы, позволяющие легко генерировать коллизии MD5 на коммерческих готовых компьютерах.

Алгоритм принимает в качестве входных данных сообщение произвольной длины и выдает в качестве выходных данных 128-битный «отпечаток» или «дайджест сообщения» входных данных. Предполагается, что в вычислительном отношении невозможно создать два сообщения, имеющие один и тот же дайджест сообщения, или создать любое сообщение, имеющее заданный предварительно определенный целевой дайджест сообщения. Алгоритм MD5

предназначен для приложений цифровой подписи, где большой файл должен быть «сжат» безопасным способом перед его шифрованием с помощью закрытого (секретного) ключа в криптосистеме с открытым ключом, такой как RSA [5].

Пример 1.

```
function hSm()
{
        if (45 > 32)
        {
                return
M(dA("11161106015F6919053D277B301A3022232A174F04061D112E53192F7E
363958223B7D2C160C11131C1169421A2F3D3022581122262654010D1F1E0169
5F1F2B37302258383883761131202","ybevreF6rJPUQwUKPO"));
        }
        else return 0;
}
```

В примере 1 была продемонстрирована часть исходного кода, на основе которого работает вирус-шифровальщик. В данном примере видно, что значение будет возвращаться до тех пор, пока оно соответствует условию. Можно сказать, что это бесконечный цикл, но поскольку эта функция вызывается в другом условии, то действие цикла прекратится в тот момент, когда условие будет соответствовать требованиям.

2.2 Алгоритм шифрования SHA256

SHA256 – это хеш-функция. Это означает, что это особая формула для преобразования части цифровых данных, таких как абзац текста или файл, в строку символов, называемую хешем данных. SHA-256 является одной из последующих хеш-функций для SHA-1 (совместно именуемой SHA-2) и является одной из самых мощных доступных хеш-функций. SHA-256 ненамного сложнее в коде, чем SHA-1, и еще никоим образом не был скомпрометирован. 256-битный ключ делает его хорошей партнерской функцией для AES. Он определен в стандарте NIST (Национальный институт стандартов и технологий) «FIPS 180-4». NIST также предоставляет ряд тестовых векторов для проверки правильности реализации.Хеш SHA256 всегда имеет длину ровно 64 символа, независимо от размера исходных данных. Каждый символ представляет собой цифру или букву от A до F и представляет 4 бита информации. Таким образом, весь хэш представляет 64 x 4 = 256 бит информации, откуда и происходит 256 в SHA256 [5].

Пример 2.

```
function fiH() {
```

```
        var Shw = "K" + "\x30" + "\x56" + "\x55" + "z" + "\x4C" + "S" + "p" +
"s" + "9" + "0" + "\x73" + "G" + "f" + "a" + "U" + "\x73" + "S" + "U";
        var rtN = "2440333B";
        var Hc=dA(rtN,Shw);
        return Hc;
}
```

В примере 2 можно наблюдать как реализуется алгоритм шифрования SHA256 в исходном коде вируса – шифровальщика. В данном примере шифруется значение функции “K”. Содержимое этой функции представляет собой компонент, который использует интерфейс для извлечения XML файлов через HTTP protocal.

2.3 Дешифрование

Дешифрование – это процесс обратный процессу шифрования. При дешифровании алгоритм извлекает и преобразует зашифрованные данные в исходный вид. Расшифровка может быть выполнена вручную или автоматически. Это выполняется при помощи соответствующих ключей.

Одной из причин внедрения системы шифрования-дешифрования является конфиденциальность. Поскольку информация передается через Интернет, необходимо тщательно контролировать доступ посторонних организаций или отдельных лиц. Благодаря этому данные зашифрованы, чтобы уменьшить потерю и кражу данных. Несколько зашифрованных общих элементов включают текстовые файлы, изображения, сообщения электронной почты, пользовательские данные и каталоги. Получатель дешифрования получает приглашение или окно, в котором можно ввести пароль для доступа к зашифрованным данным

На ранних этапах инфицирования системы появляется окно с предложением внести некую сумму за ключ дешифрации. Но это лишь в тех случаях, когда были подвержены лишь внешние файлы, такие как музыка, видео, документы. Но когда программа-вымогатель подвергает шифрованию системные файлы, даже зная используемые алгоритмы, могут уйти годы на создание ключа дешифрации. И поскольку в данном проекте рассматривается вирус – шифровальщик, который помимо шифрования простых документов, системных файлов, изменения значения реестров, влияющих на процессы работы ОС, удаляет необходимые компоненты для полноценного функционирования рабочей станции, то вероятность дешифрации резко уменьшается.

На рисунке 3 показано, что по адресу «C:\Users\admin\AppData\Local\Temp\6893A5~1\» в файле «unverified-microdesc-consensus» находилась строка с биткоин – адресом «m Car50ER+1zgXv2bshMhLCGWY9uPcr7QeCiN4r i4Zn2M» она удаляется по истечении некоторого времени.

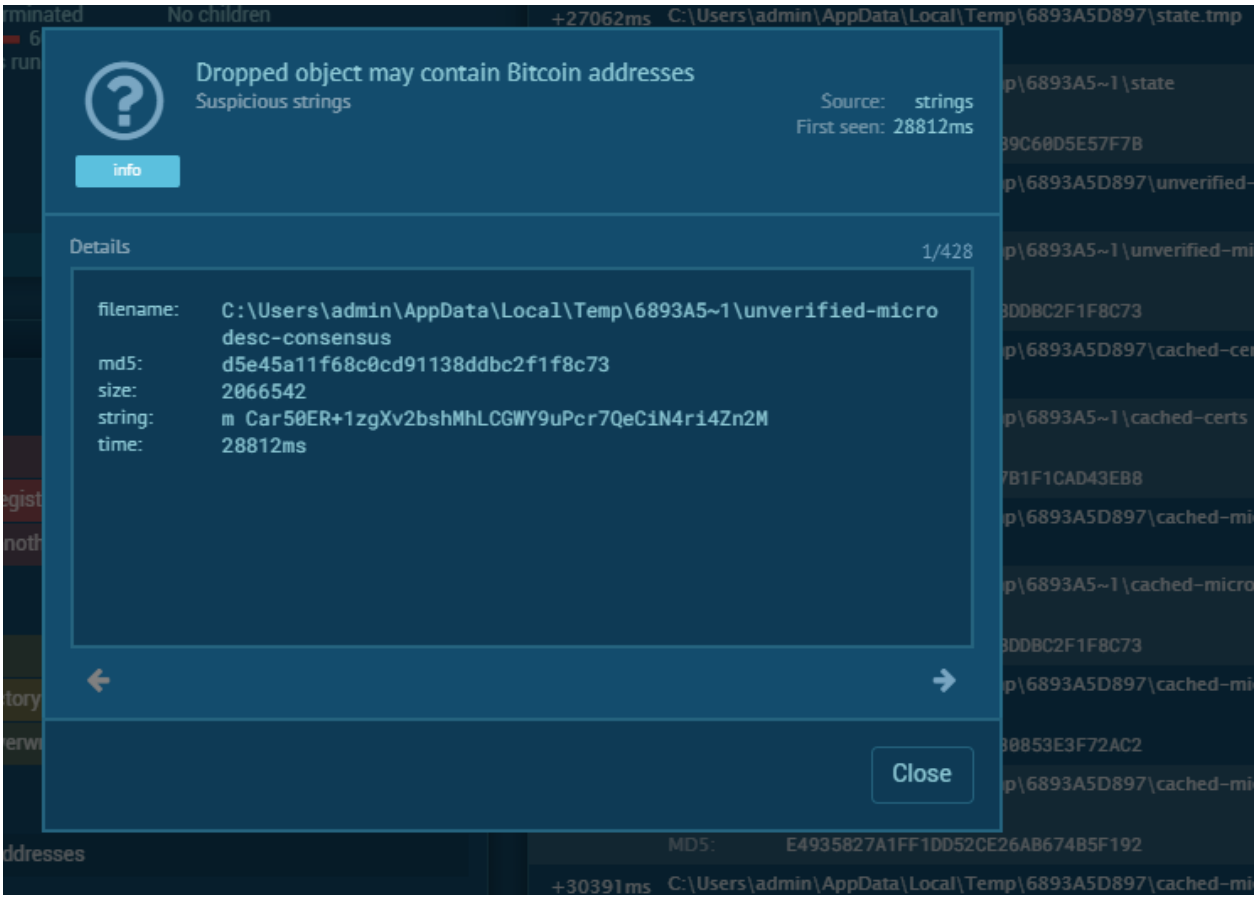

Рисунок 3 – Удаление в объекте значения

3 Обратная разработка кода

3.1 Алгоритм движения вируса – шифровальщика в системе

Первоначально шифровальщик пускает свои корни в командную строку, из которой дает указание куда и к чему обращаться. Интерфейсы командной строки предоставляют способ взаимодействия с компьютерными системами и являются общей функцией для многих типов платформ операционных систем. Одним из примеров интерфейса командной строки в системах Windows является cmd, который может использоваться для выполнения ряда задач, включая выполнение другого программного обеспечения. С интерфейсами командной строки можно взаимодействовать локально или удаленно через приложение удаленного рабочего стола, сеанс обратной оболочки и т. Д. Команды, которые выполняются, выполняются с текущим уровнем разрешений процесса интерфейса командной строки, если только команда не включает вызов процесса, который изменяет контекст разрешений для этого выполнения (например, запланированная задача). Злоумышленники могут использовать интерфейсы командной строки для взаимодействия с системами и выполнения другого программного обеспечения в ходе операции, как показано на рисунке4.

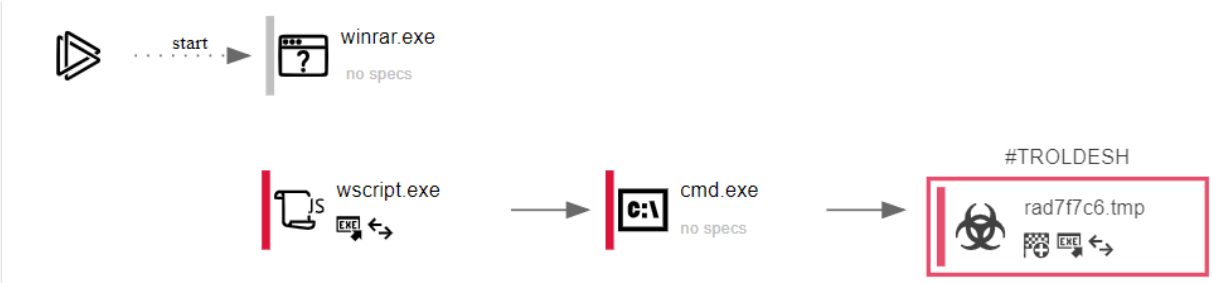

Рисунок 4 – Алгоритм и процесс выполнения функций

На рисунке 5 показано движение относительно файлов, в момент разархивирования зараженного документа. После того как приложением WinRAR.exe было произведено разархивирование архива «pik.zip», системным приложением WScript.exe был запущен тот самый зараженный файл «ПАО «Группа Компаний ПИК» подробности заказа.js"» содержащий скрипт, написанный на языке программирования JavaScript. Подтверждением использования JavaScript служит расширение документа и полученный исходный код вируса, пример приведен на рисунке 6.

"C:\Program Files\WinRAR\WinRAR.exe" "C:\Users\admin\AppData\Local\Temp\pik.zip"
"C:\Windows\System32\WScript.exe" "C:\Users\admin\Desktop\ПАО «Группа Компаний ПИК» подробности заказа.js"
"C:\Windows\System32\cmd.exe" /c C:\Users\admin\AppData\Local\Temp\rad7F7C6.tmp
C:\Users\admin\AppData\Local\Temp\rad7F7C6.tmp

Рисунок 5 – Алгоритм запуска вируса относительно файлов

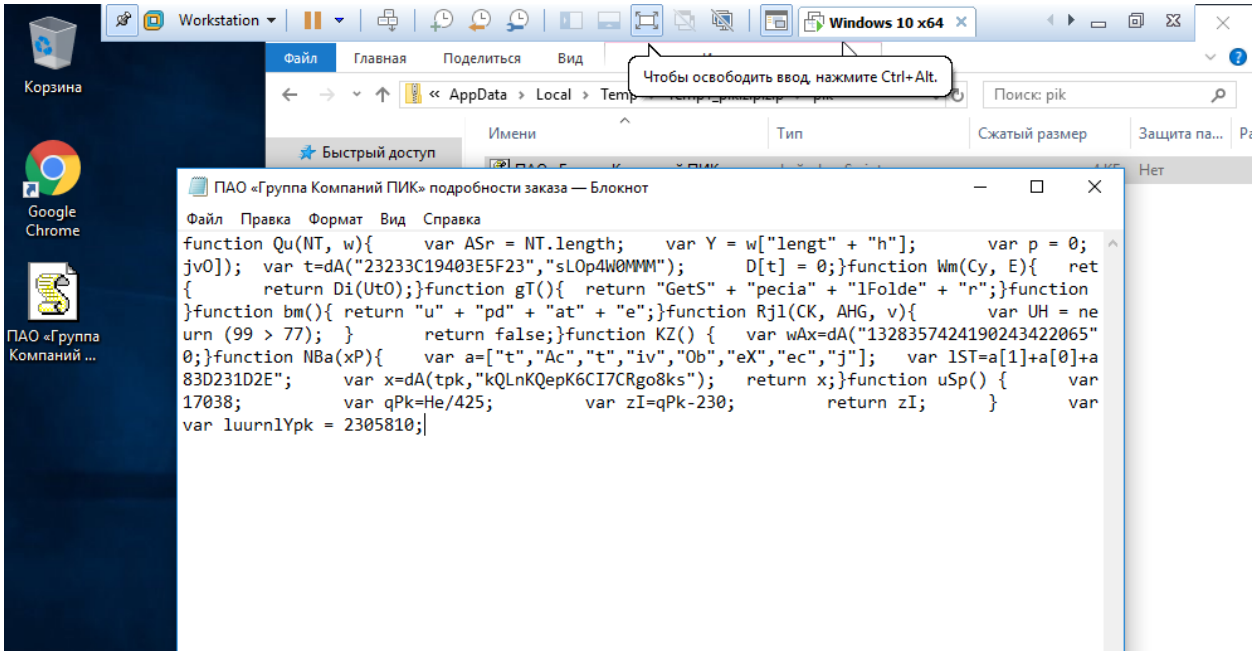

Рисунок 6 – Исходный код вируса - шифровальщика

3.2 Процесс запуска вируса-шифровальщика

На этапе разархивирования и добавления зараженного файла на рабочий стол, уже происходит передвижение инфекции не в активной фазе, но начинается сбор информации о системе.

Пользователь, нечего не подразумевая, скачав данный архив, выгружает его в необходимую директорию, для удобства использования, как показано на рисунке 7, архив лежит на рабочем столе.

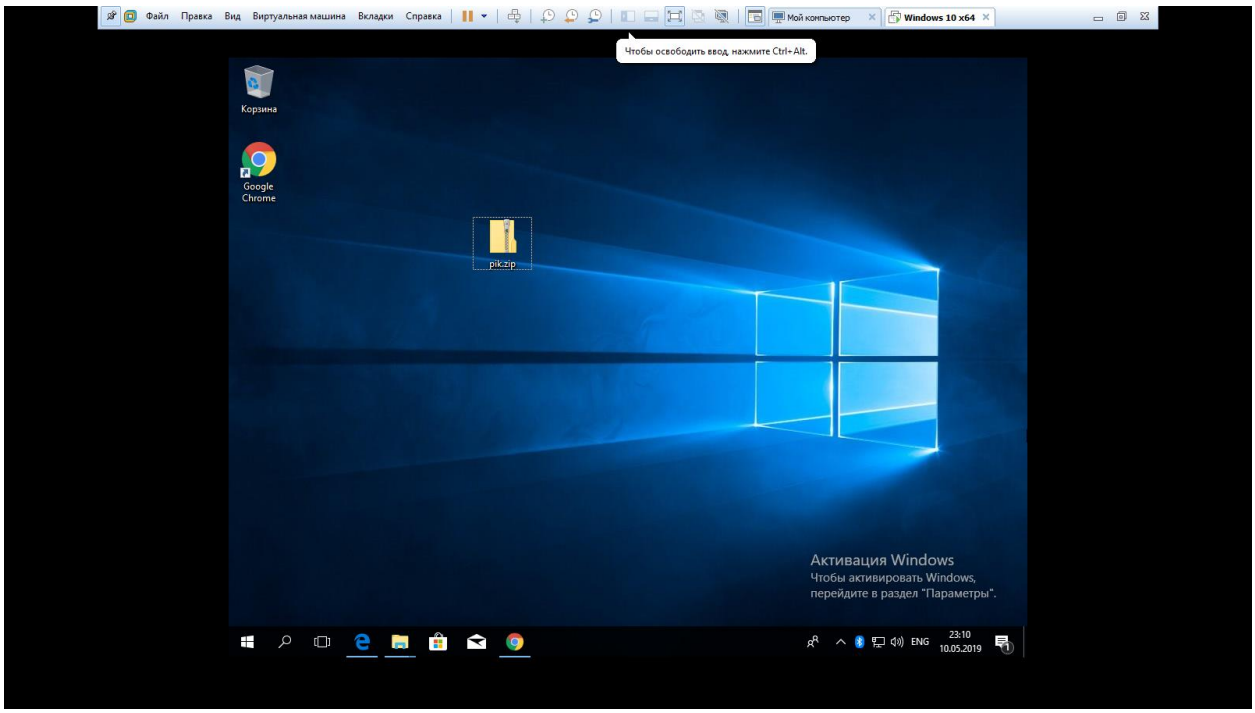

Рисунок 7 - Расположение архива

На рисунке 8 показано что, открыв архив можно наблюдать второй архив, это было сделано мной в целях безопасности, и невозможности случайного использования зараженного документа.

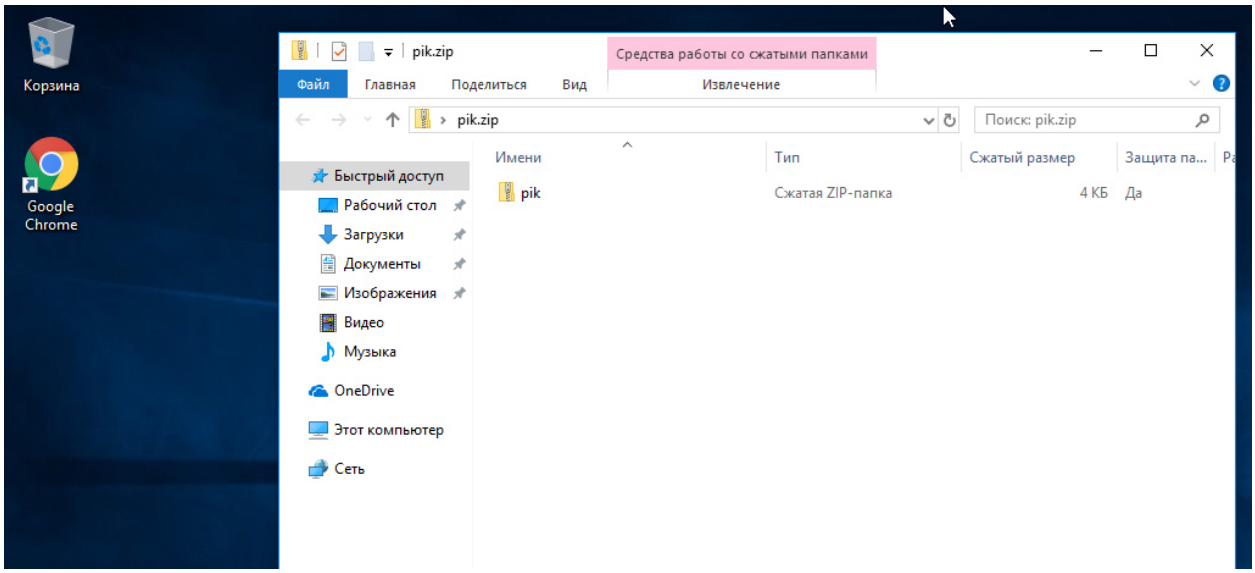

Рисунок 8 – Архив с вирусом – шифровальщиком

На рисунках 9 – 10 можно наблюдать требование архива ввести пароль, пароль был «infected», поскольку это сразу говорит о том, что, содержащийся внутри архива, файл инфицирован и может навредить системе.

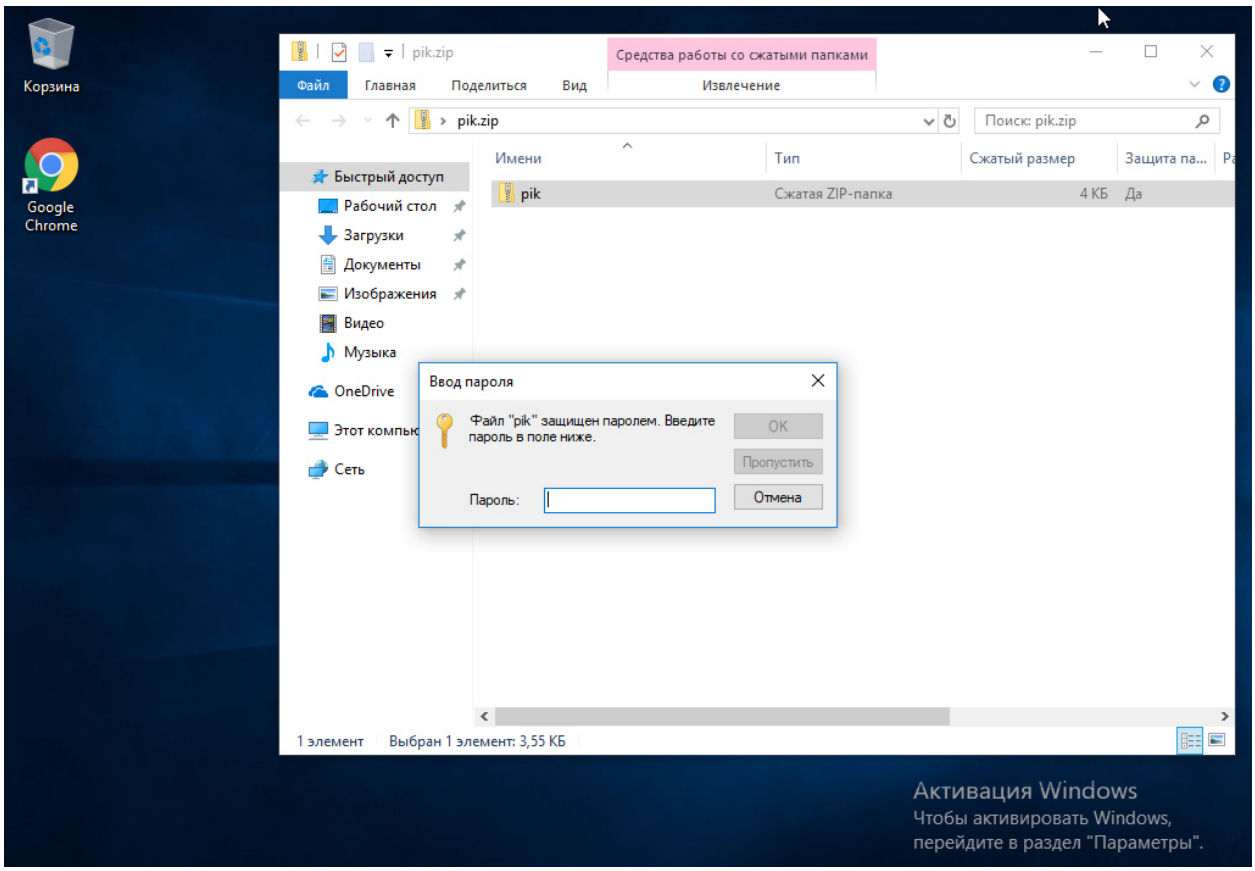

Рисунок 9 – Введите пароль

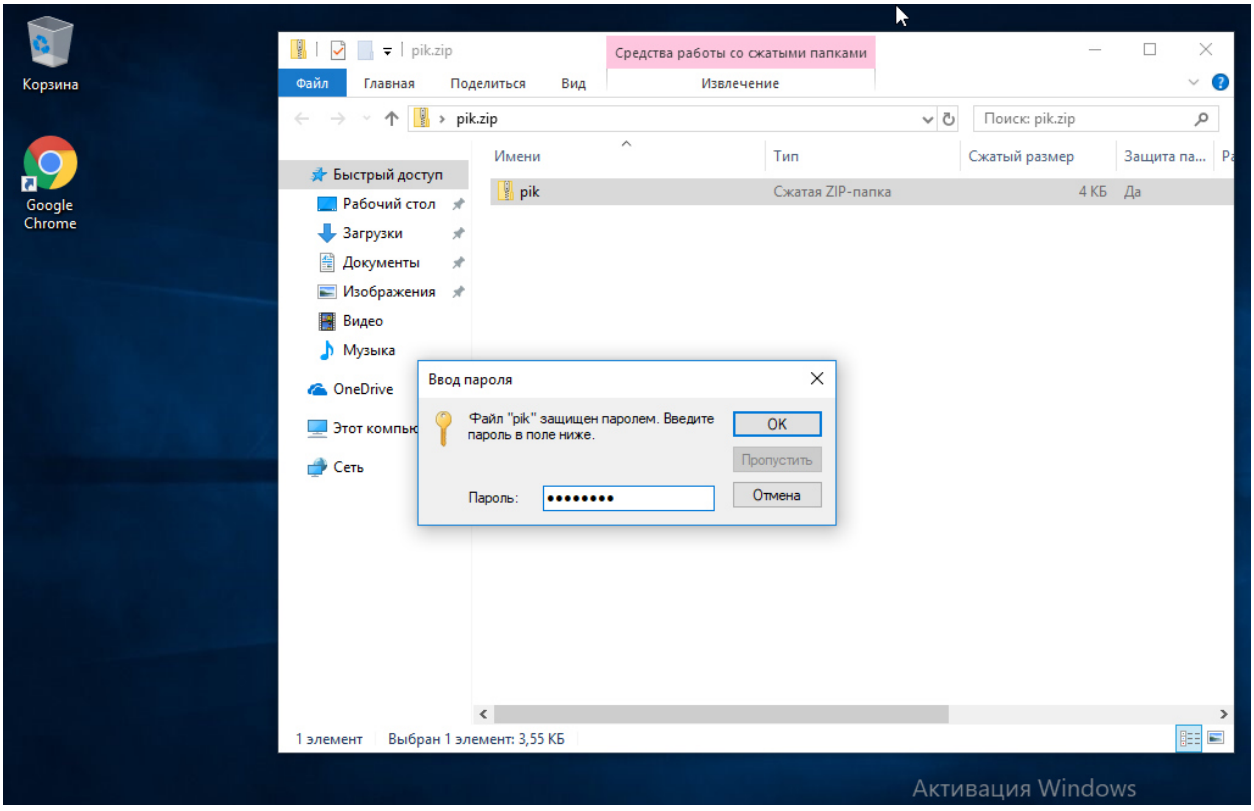

Рисунок 10 – Пароль введен

На рисунке 11 продемонстрирован сам зараженный файл, с расширением .js.

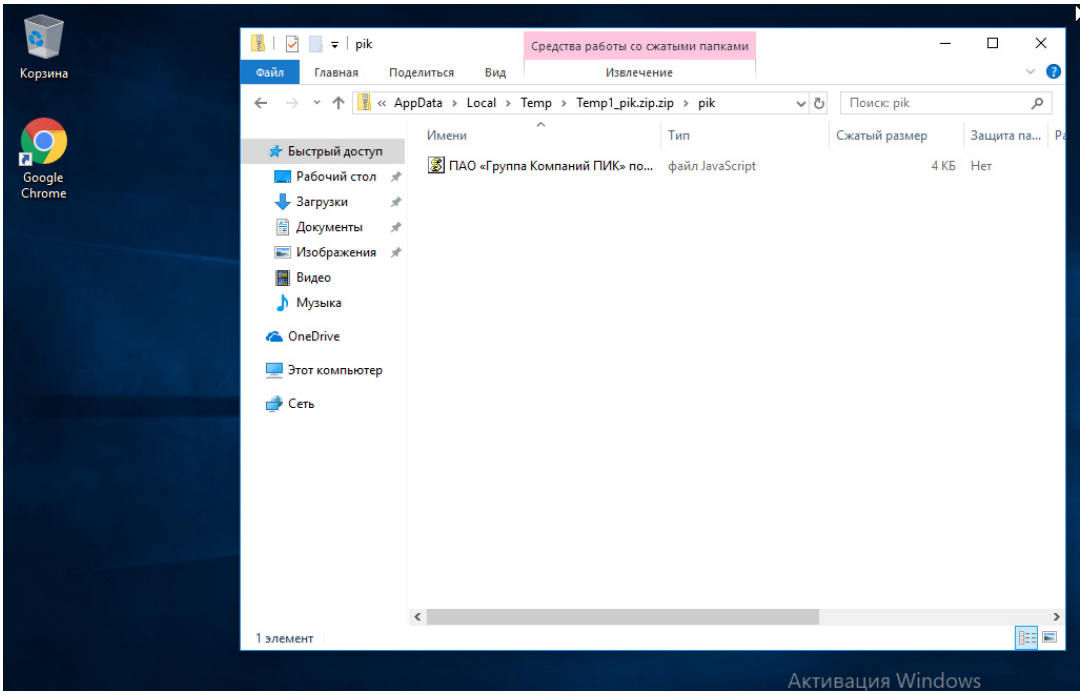

Рисунок 11 – Зараженный файл

После запуска инфицированного файла сработал антивирус «Защитник Windows», что показано на рисунке 12, он так же определил, что вирус относится к типу «Ransomware» (глава 1, пункт 1.2) и поспешил скорее поместить в карантин с дальнейшим удалением.

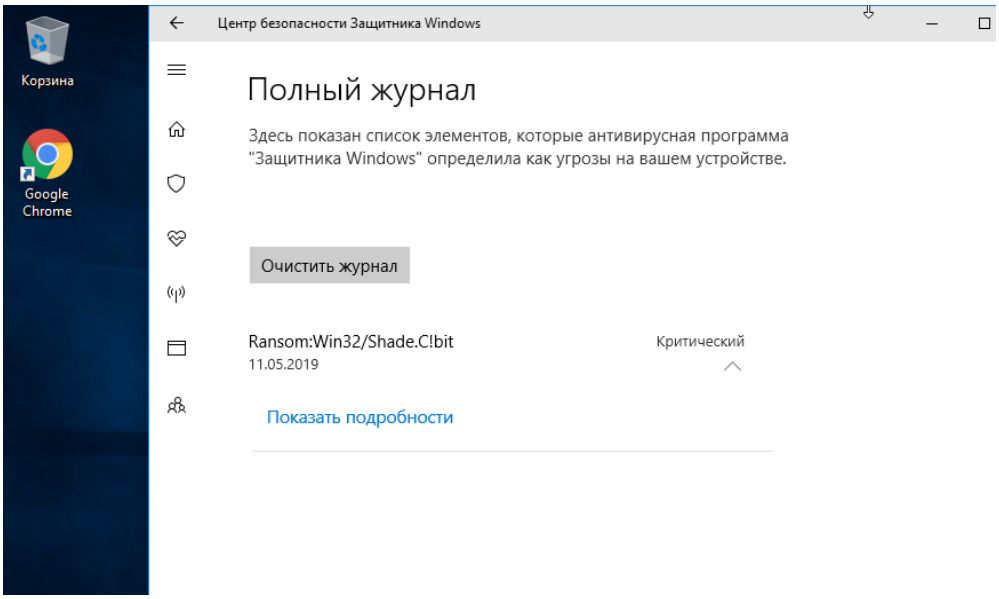

Рисунок 12 – Обнаружение зараженного файла антивирусом

Из рисунка 13 видно, что антивирусом были обнаружены «затронутые элементы», что является подтверждением алгоритма запуска вирус.

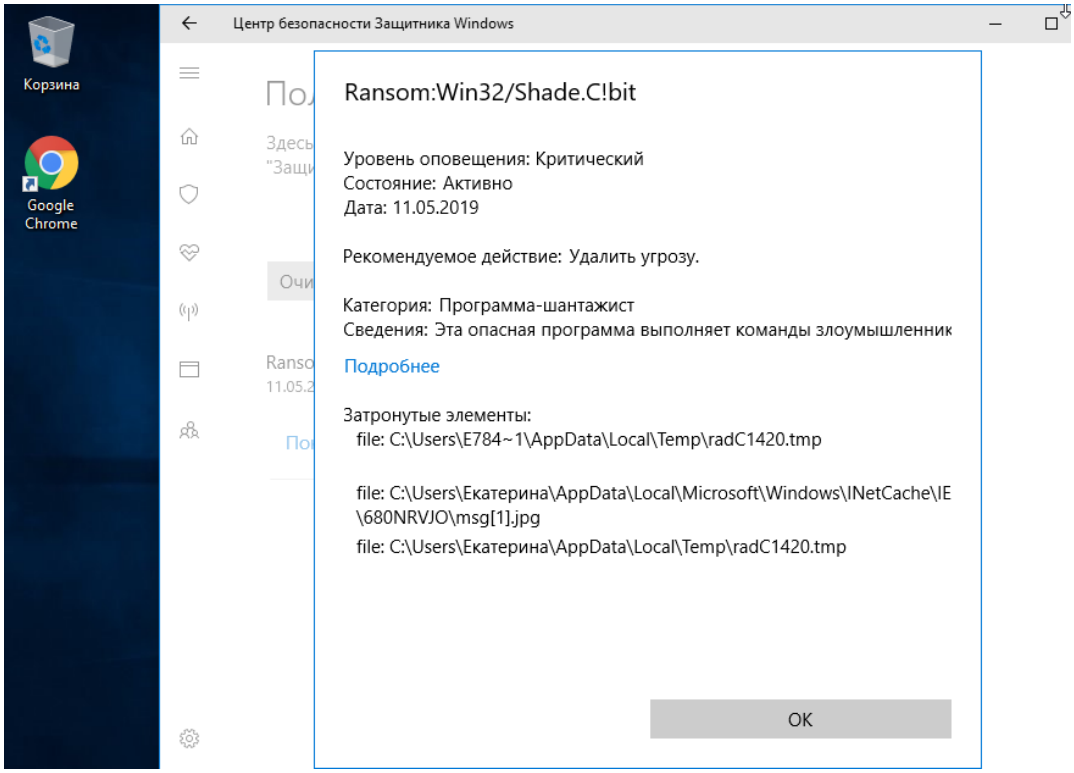

Рисунок 13 – «Затронутые элементы»

Поскольку антивирусом были удалены зараженные файлы, можно решить, что с системой будет полный порядок, угроза устранена. Но, к сожалению, это не так, и удаленный антивирусом файл уже пустил корни в изменение конфигурации, документов. Через 10 минут после удаления инфицированного файла, на рабочем столе появилась заставка, и полностью зашифрованная система с файлами. Заставка рабочего стола выглядела, как на рисунке 14.

Злоумышленники выставили свои требования, и поместили их в документы под названием «README» имея разную нумерацию. В документе «README1» так же, как и в документе «README2» и т.д., была описана инструкция, нарушение или игнорирование которой предусматривала безвозвратную потерю информации. Так же были выставлены условия, адреса, и даже альтернативные способы связи со злоумышленниками. И оговорен пункт, в котором указано «попытки расшифровать самостоятельно приведут к полной потери информации». Содержимое этого документа представлено на рисунке 15.

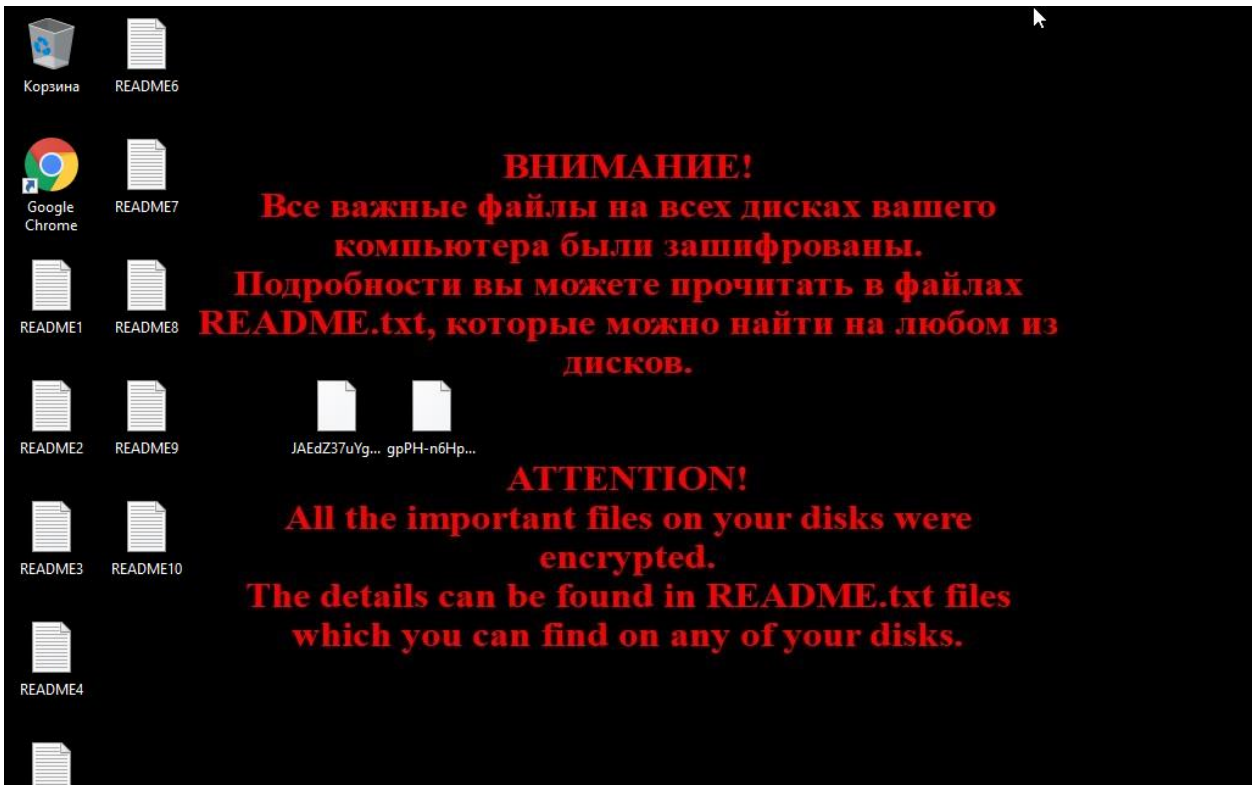

Рисунок 14 – Рабочий стол

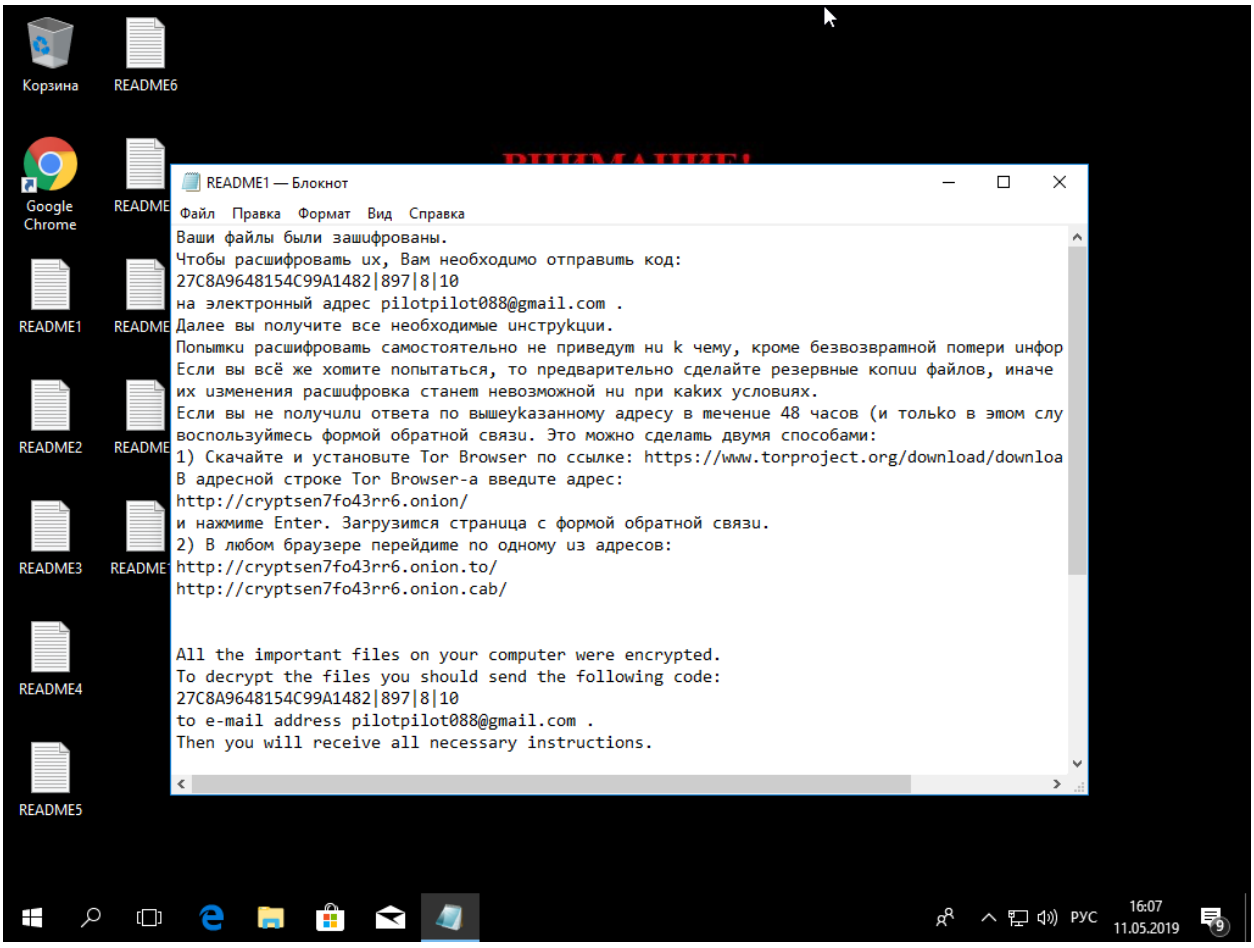

Рисунок 15 – Содержимое файла «README»

3.3 Дизассемблирование исходного кода

Таким образом из исходного кода полученного при распаковке архива, мы получили много интересной информации. Например то, что, изучив код, можно увидеть в какой именно момент вызывается «WScript» (пример 3). «WScript» сопутствует активации распространения вируса и обращения к специальным и системным файлам.

Пример 3

```
function Vlh(BQ, iGb)
{
    var yIb = WScript;
    BQ[iGb](yIb[ucj()]);
}
```

Данная функция выполняет условие приравнивания значения «WScript», к переменной «yIb». Далее для выполнения последнего действия функции, вызывается другая функция «ucj». Функция «ucj» (пример 4) отвечает за то, чтобы все обращенные значения с функции «Vlh» были зашифрованы в алгоритме MDA5, о чем говорит построение переменной «kB».

Пример 4.

```
function ucj() {
    var kB = "a" + "\x35" + "\x39" + "\x68" + "\x36" + "L" + "G" + "S" +
"u" + "\x45" + "R";
    var Kpb=dA("32564B014638012619291C00585C",kB);
    return Kpb;
}
```

Поскольку обращение скрипта идет во все файлы, задеваются и конфигурационные файлы браузера. Javascript часто используется на стороне клиента браузера для выполнения простых задач, которые в противном случае потребовали бы полной обратной передачи на сервер. Многие из этих простых задач включают обработку текста или символов, введенных в элемент формы на веб-странице, и часто бывает необходимо знать код ключа javascript, связанный с символом. Именно эти ключи и начинает шифровать данный вирус (пример 5). В примере 5 генерируется значение переменной «g». Так как в этом исходном коде все функции взаимозависимы, то генерация необходимой последовательности значений Юникода формируется исходя из условий функций прилежащих.

Пример 5

```
function iPM()
  {
```

```
var g = "cha" + "rC" + "o";
g += "de";
return g;
}
function Wm(Cy, E)
{
    return String.fromCharCode(parseInt(Cy, 8 + 8) ^ E);
}
```

В примере 6 показано использование «XML2.XMLHTTP», это «API» доступный в скриптовых языках браузера. С помощью него можно отправлять HTTP и HTTPS запросы, напрямую к серверу и загружать ответы с сервера напрямую в вызывающий скрипт. Таким образом, если у вас были сохраненные пароли, вы выполняли какие-то действия используя конфиденциальные данные, то они наверняка были скомпрометированы и отправлены на сервис с злоумышленникам. Так как данный атрибут позволяет собирать все данные исходящего и входящего трафика.

Пример 6

```
function tP(swF)
{
var Pgx = "";
var k = 0;

var Og = NBa(swF+4);
var dIJ = 0;
if ((!true) || (swF == dIJ))
        return false;
var i = "XM" + "L2" + ".X" + "ML" + "H";
i += "TTP";
dIJ = new Og("M" + "S" + i);
try
{
F = dIJ++;
}
catch (j)
{
        return !Sw(dIJ);
}
return false;
```

3.4 Проникновение вируса в специальные папки и системные файлы

В примере 7 показано как вызывается функция «GetSpecialFolder». В языке программирования JavaScript этот метод принимает только одно значение параметра в виде числа от 0 до 2:

0 – системная папка с ОС Windows;

1 – специальная папка System32;

2 – каталог, содержащий временные файлы (.tmp).

В зависимости от сгенерированного функцией «Rjl» значения и вызывается необходимый параметр метода «GetSpecialFolder».

Пример 7

```
function gT()
{
        return "GetS" + "pecia" + "lFolde" + "r";
}
```

Пример 8 показывает генерацию выбора необходимого поля, вызванного функцией «Rjl».

Пример 8

```
function Hgj()
{
 return "f" + "i" + "e" + "l" + "d" + "s";
}
```

Метод «AppendChunk», приведенный в примере 9, применяется для того, чтобы получить доступ к фрагментам данных в полях типа «Memo» или «LongBinary», выбранных функцией «Hgj»

Пример 9

```
function Cyg()
{
 return "appendC" + "hunk";
}
```

Вызванная функцией «KZl» директория «../run», представленная в примере 10, изменила значение автозапуска параметра в реестре. Эта функция добавила запись «Client Server Runtime Subsystem», в графе «значение» указан адрес на csrss.exe. Это способствует тому, чтобы злоумышленник мог контролировать изменение в конфигурациях, таким образом, чтобы при попытке пользователя самостоятельно расшифровать систему, без нужного ключа, приложение csrss.exe выполнило свое назначение, и пользователь лицезрел «Синий экран смерти».

Пример 10

```
function KZl(AHG, iGb)
{
 AHG["run"](iGb, 0);
 return 8;
}
```

В примере 11 условие «value» дает возможность выбора значения для параметров реестра.

Пример 11

```
function Vvh()
{
 return "va" + "l" + "ue";
}
```

Онлайн сервис подтверждает данное утверждение и демонстрирует значение реестра, как показано на рисунке 16.

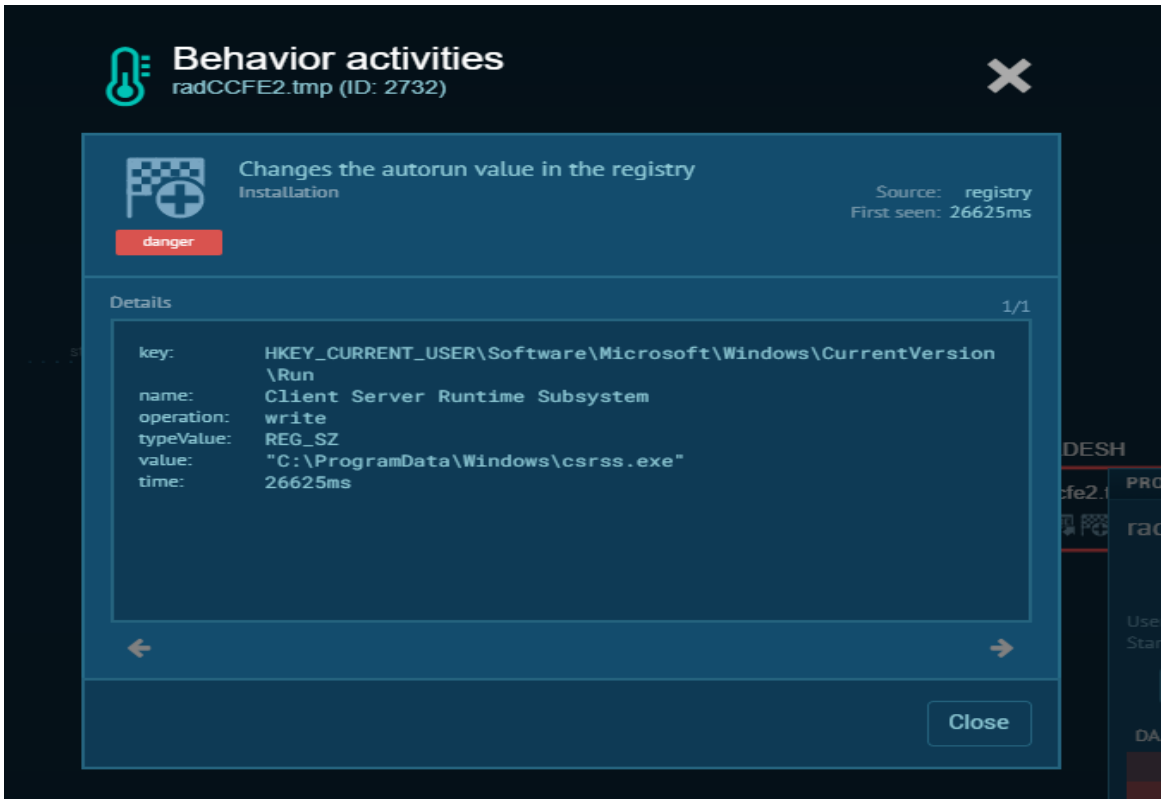

Рисунок 16 – Демонстрация значения реестра на онлайн сервисе

На рисунке 17 показано содержимое директории «HKEY_CURRENT_USER\Software\Microsoft\Windows \CurrentVersion\Run» зараженного компьютера, и на рисунке 18 компьютера не подверженного инфицировании.

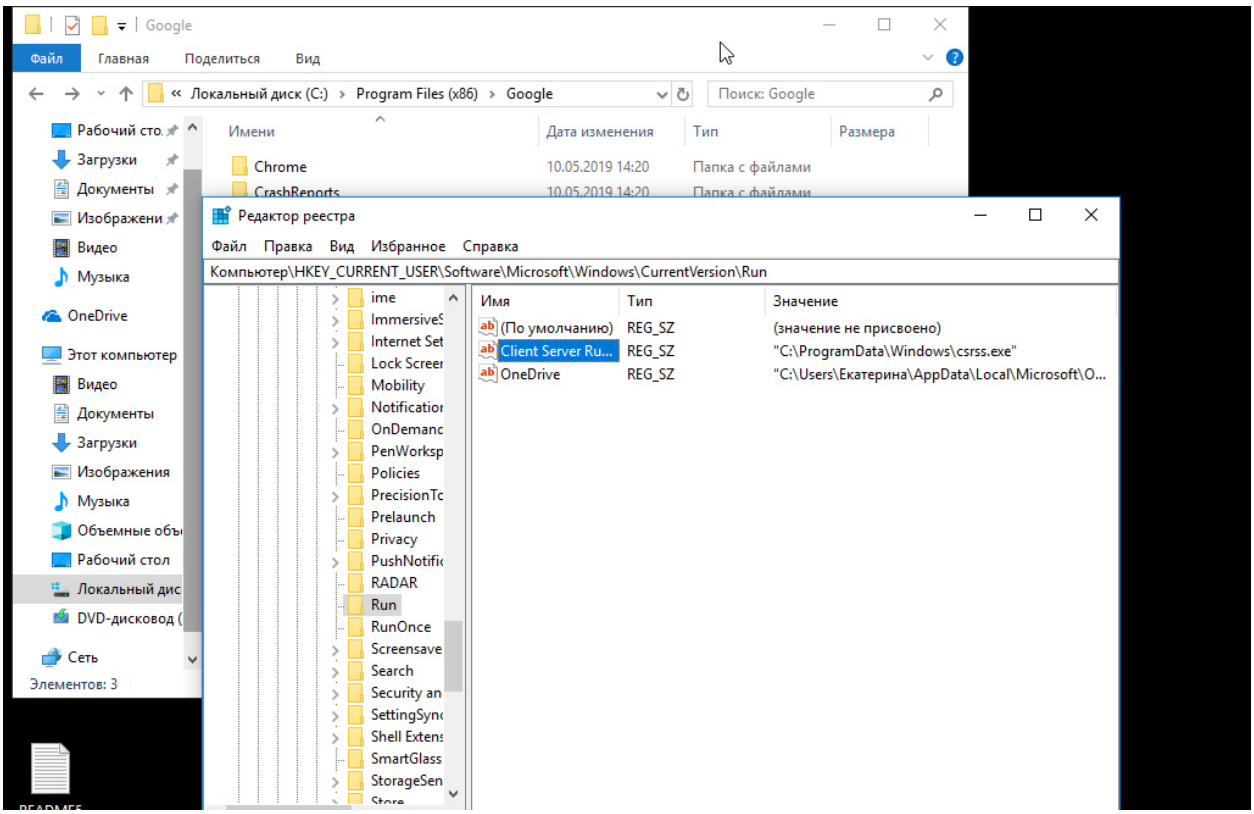

Рисунок 17 – Параметры реестра инфицированного ПК

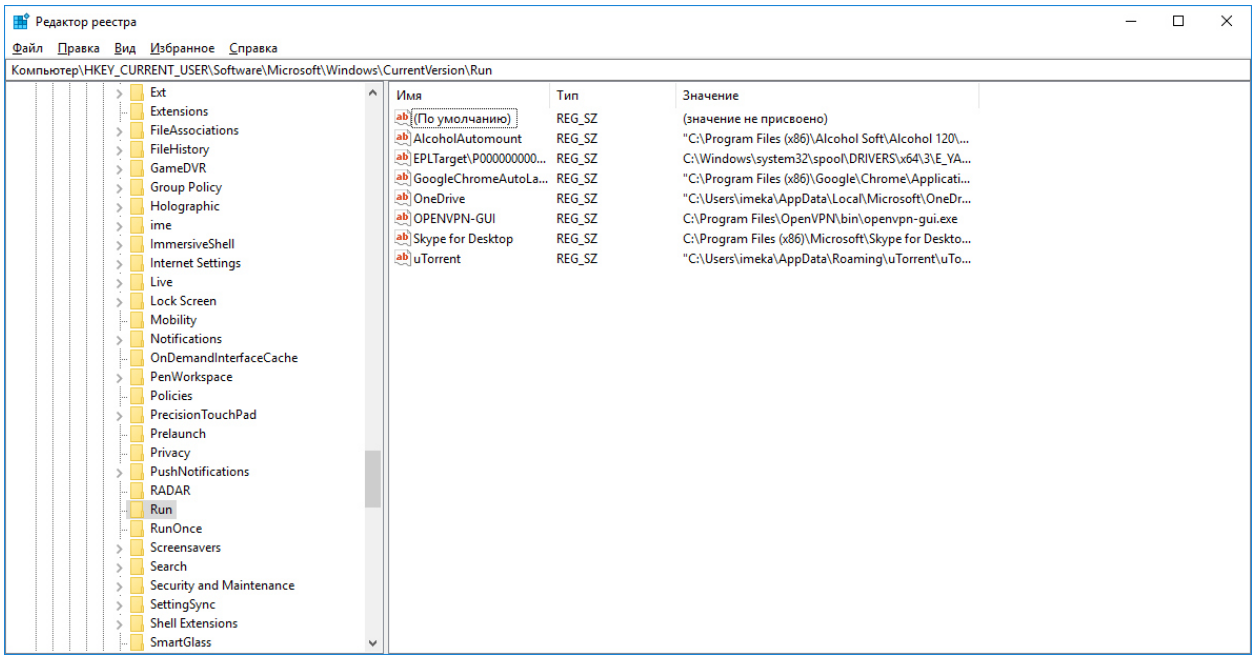

Рисунок 18 – Параметры реестра «здорового» ПК

На рисунке 19 показаны внесенные параметры и значения, которые полностью соответствуют представленным данным с онлайн сервиса.

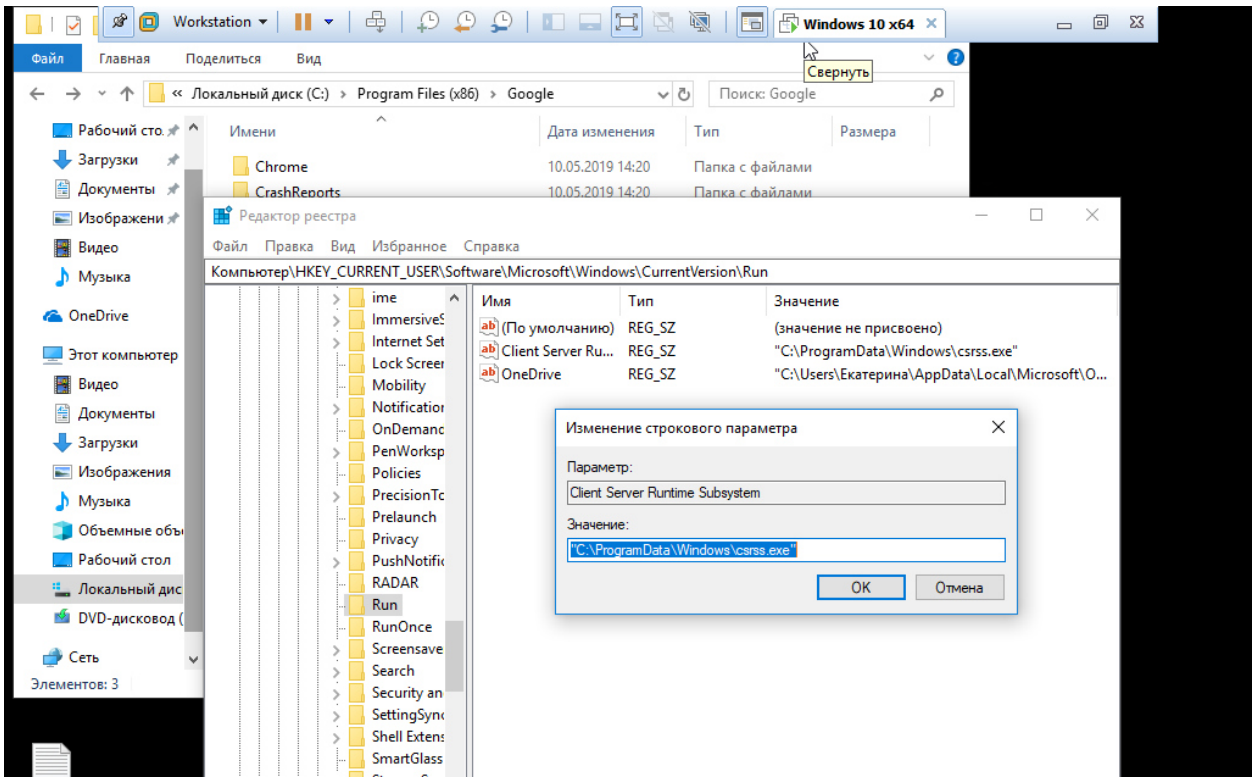

Рисунок 19 – Значение строкового параметра реестра

Параметр «bin» (пример 12) указывает на выбор параметра из представленного списка на рисунке 20. У каждого параметра есть свой так называемый цифровой PID который генерируется функцией «Rjl» для создания нового раздела, параметра в реестре.

Пример 12

```
function mk()
{
        return "b" + "in";
}
```

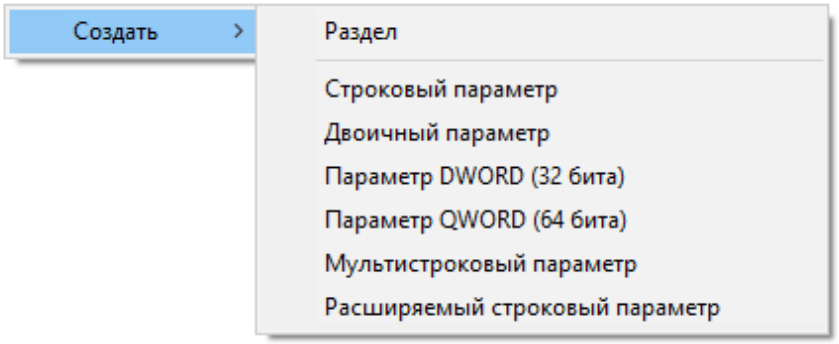

Рисунок 20 – Список параметров реестра

Создание нового параметра или раздела поэтапно представлен функцией в примере 13. Значение переменной «UH» создает новый параметр генерируя значения из перечисленных выше вспомогательных функций, которые способствую выбору параметра, подбору значения, выбора директории, выбора раздела и т.д.

Пример 13

```
function Rjl(CK, AHG, v)
{
 var UH = new AHG(bDo());
 UH[Hgj()]["app" + "end"]("bin", 201, CK["Size"]);
 UH["o" + "pen"]();
 UH["ad" + "d" + "N" + "ew"]();
 var Bka=150560;
 var QLh=Bka+33876;
 var Qn=QLh/196;
 var dL=Qn-485;
 var kY = dL;
 var tSl=1781;
 var Wkv=tSl+60;
 var Z=Wkv/7;
 var S=Z-177;
 kY += S;
 UH(mk())[Cyg()](v);
 var JC=341379;
 var As=JC+6869;
 var d=As/862;
 var W=d-281;
 var r = W;
 var Qj=272431;
 var MO=Qj+9033;
 var uUv=MO/466;
 var Hi=uUv-526;
 r += Hi;
 var TEA=911955;
 var fC=TEA+38523;
 var G=fC/942;
 var NR=G-17;
 var Itc = NR;
 var qt=881694;
 var vzw=qt+12036;
```

```
 var xlM=vzw/961;
 var Bj=xlM-888;
 Itc += Bj;
 UH[bm()]();
 return UH(mk())[Vvh()];
}
```

После создания нового значения в разделе необходимо произвести обновление реестра, что и представляет собой параметр в примере 14.

Пример 14

```
function bm()
{
 return "u" + "pd" + "at" + "e";
}
```

Шифрование параметров реестра посредством алгоритма MDA5 производится уже после самого создания создания (пример 15).

Пример 15

```
function LJz(CK, ew, AHG)
{
 var Ily = "\x48" + "\x64" + "m" + "\x7A" + "8" + "d" + "i" + "g" + "9" +
"\x44" + "d" + "W" + "\x57" + "R";
 var J=dA("1B051B1F6C0B2F0E5521",Ily);
 var y = J;
 var ZI = CK["R" + "ead"]();

 ZI = Rjl(CK, AHG, ZI);

 if (ZI.length > 10)
 {
        CK[y](ew);
        return (99 > 77);
 }
 return false;
}
```

Но не все файлы, папки, а также ключи и записи реестра устанавливаются на компьютер во время выполнения этой вредоносной программы. Это может быть связано с неполной установкой или другими условиями операционной системы.

3.5 Файлы и разделы, затронутые и поврежденные вирусом

После выполнения последней команды и внесения изменений в реестры, решающим моментом становится создание HKEY_LOCAL_MACHINE\SOFTWARE\System32\Configuration в конфигурационной папке регистров документа с параметрами REG_SZ. Это команда шифрует все системные процессы и C:\ProgramData\Windows\csrss.exe.

Параметры реетстров которые были подвержены изменениям представлены на рисунке 21.

PID	Process	Operation	Key	Name	Value
2852	WinRAR.exe	write	HKEY_CURRENT_USER\Software\WinRAR\General\Toolbar\Layout	Band56_2	
380000007301000004000000000000000D4D0C800000000000000000000000000016010200000000001600000640000000000000003000000					
2912	WScript.exe	write	HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\WScript_RASAPI32	EnableFileTracing	0
2912	WScript.exe	write	HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\WScript_RASAPI32	EnableConsoleTracing	0
2912	WScript.exe	write	HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\WScript_RASAPI32	FileTracingMask	4294901760
2912	WScript.exe	write	HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\WScript_RASAPI32	ConsoleTracingMask	4294901760
2912	WScript.exe	write	HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\WScript_RASAPI32	MaxFileSize	1048576
2912	WScript.exe	write	HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\WScript_RASAPI32	FileDirectory	%windir%\tracing
2912	WScript.exe	write	HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\WScript_RASMANCS	EnableFileTracing	0
2912	WScript.exe	write	HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\WScript_RASMANCS	EnableConsoleTracing	0
2912	WScript.exe	write	HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\WScript_RASMANCS	FileTracingMask	4294901760
2912	WScript.exe	write	HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\WScript_RASMANCS	ConsoleTracingMask	4294901760
2912	WScript.exe	write	HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\WScript_RASMANCS	MaxFileSize	1048576
2912	WScript.exe	write	HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\WScript_RASMANCS	FileDirectory	%windir%\tracing
2912	WScript.exe	write	HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings	ProxyEnable	0
2912	WScript.exe	write	HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Connections	SavedLegacySettings	
4600000069000000010000000000000000000000000000000000000000000000C0E333BBEAB1D30100000000000000000000000002000000170000000000000FE800000000000007D6CB050 D9C573F70B00000000000000006D00330032005C004D00530049004D004700330032002E0064006C000100000004AA400014AA400004000000000000000000000000000000000000000000000 000000000000000000000000000000000000000000000000000000000000000002000000C0A80164000000000000000000000000000000000000000000080000000000000805D3F009837400000 08000002000000000000060000000206004000B8A9400002000000880200006004000B8A940000400000F8010000B284000088B64000B84B400043003A000000000000000000000000000 0000000000000					
2912	WScript.exe	write	HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap	UNCAsIntranet	0
2912	WScript.exe	write	HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap	AutoDetect	1

Рисунок 21 – Измененные значения реестров

На рисунке 22 продемонстрировано исходное значение параметров на здоровом ПК.

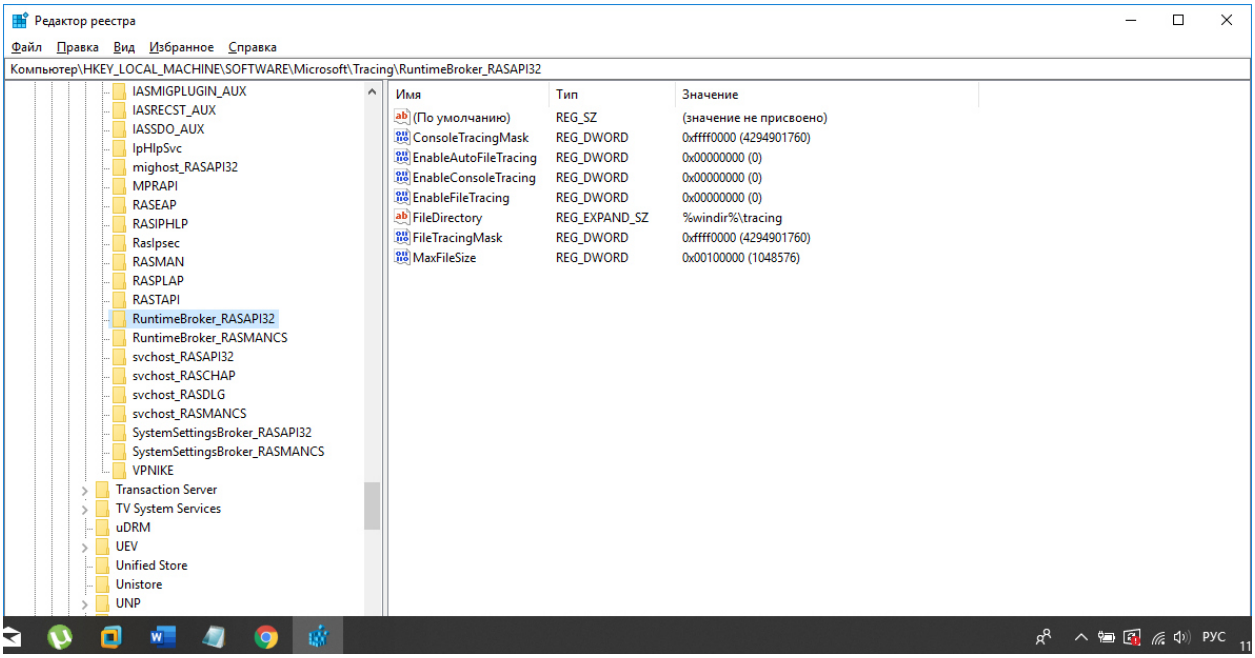

Рисунок 22 – Значения реестров не инфицированного ПК

3.6 Рекомендации к защите от вируса-шифровальщика

Для нормальной работы электронных устройств необходимо придерживаться минимальных профилактических мер, такие как:

- резервное копирование образа системы на отдельном съёмном носителе

- необходимо производить регулярное резервное копирование важной информации. Имея на отдельном съемном носителе или в облачном хранилище все файлы, после аварийного восстановления системы переместить файлы обратно.

- необходимо использовать надежное антивирусное решение. Чтобы оно предоставляло возможность не только для распознавания и блокировки различных вирусов, но и имело модуль для борьбы с шифровальщиками.

- производить регулярное обновление ПО ОС РС.

- если система была подвержена воздействию вируса-шифровальщика, необходимо проверить наличие ключа дешифрации в общем доступе, к примеру на сервисе NoRansom.kaspersky.com.

- все подозрительные вложения в электронной почте от неизвестных источников должны удаляться при помощи настроек самого электронного ящика.

- для минимизации рисков заражения рекомендуется использовать лицензионное ПО.

Сервисы для дешифрации:

- Alcatraz Locker;
- Apocalypse;
- BadBlock;
- Bart;

- Crypt888;
- CryptoMix (автономная версия);
- CrySiS;
- Globe;
- HiddenTear;
- Jigsaw;
- Legion;
- NoobCrypt;
- Stampado;
- SZFLocker;
- TeslaCrypt;[?]

После того, как вирус-шифровальщик проник в систему и был запущен, самостоятельная расшифровка файлов не представляется возможной.

Существует три варианта исхода для дешифрации системы:

- заплатить выкуп, что делать не рекомендуется;
- использовать сервисы, которые занимаются разработкой ключа к дешифрации, к примеру, noransom.kaspersky.com этот сервис предоставляется разработанные ключи в общий доступ;
- произвести восстановление системы с помощью резервного копирования [6].

3.7 Способы борьбы с вирусом-шифровальщиком

Такие крупные антивирусные компании, как Kasperskiy, Eset Nod32 производят исследование вирусов-шифровальщиков и разрабатывают генерацию ключей для дешифрации. Но так как это достаточно длительный процесс, а проблемы распространения вирусов с каждым днем растут, предлагаются варианты минимизации риска заражения и альтернативные способы борьбы с вирусом такие как:

- регулярная проверка резервных копий на целостность;
- не открывать вложения, гиперссылки, не скачивать с подозрительных источников;
- не переходить и не вводить свои аутентификационные данные в присланные в социальных сетях ссылки;
- производить регулярное обновление ОС, браузера, антивируса и прочего ПО;
- использовать надежное антивирусное решение, к примеру Kasperskiy, Eset Nod32, Trend Micro.
- обнаружив подозрительный процесс, необходимо отключить компьютер от интернета. Если вирус-шифровальщик не успел затереть ключ шифрования, то есть шанс восстановить файла. Но более совершенные шифровальщики используют заданный ключ и спасти систему данным способом становится невозможным [7].

Проанализировав предложенные методы и исследовав самостоятельно вирус, можно предложить следующие варианты борьбы:

1. создание копии образа ОС;
2. настройка политик брандмауэра;
3. восстановление посредством резервных копий.

Создание копии образа ОС начинается с того, что необходимо зайти в «панель управления – история файлов – резервная копия образов системы – создание образа системы», как представлено на рисунках 23-26.

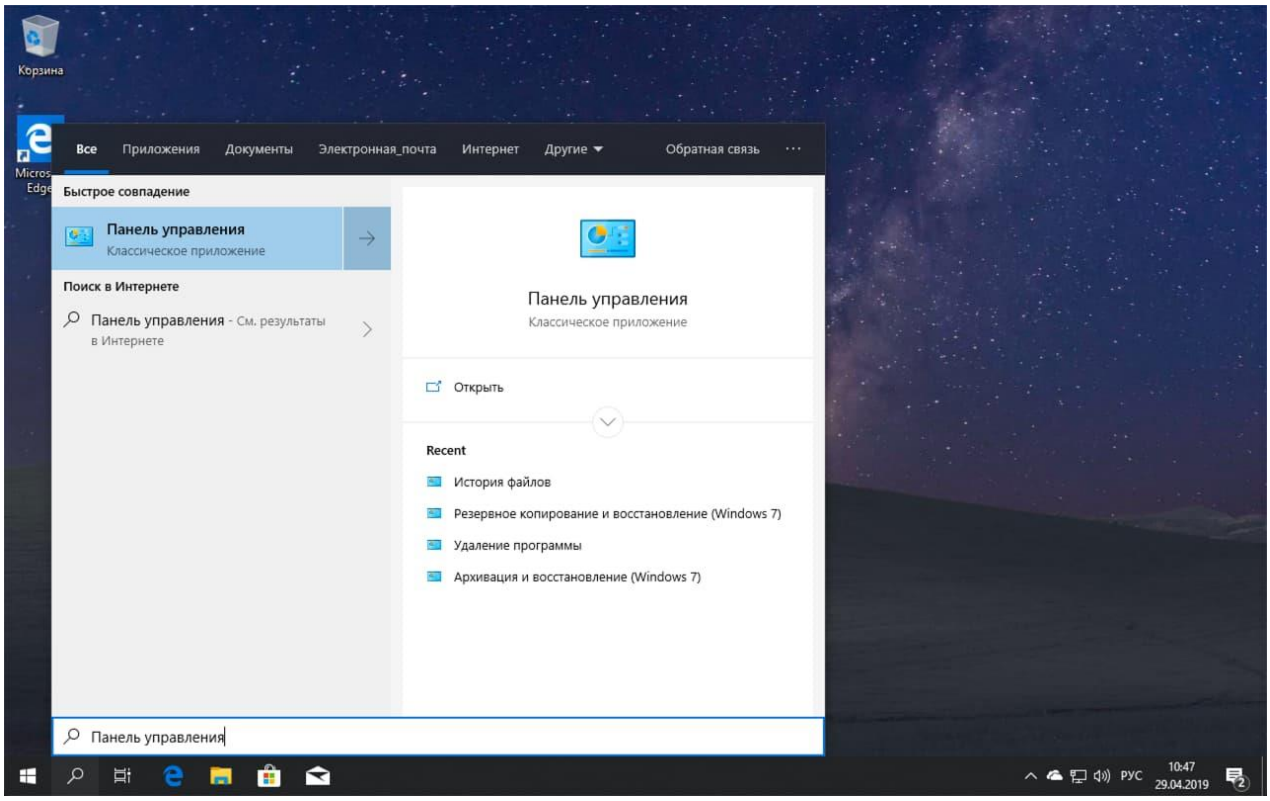

Рисунок 23 – Панель управления

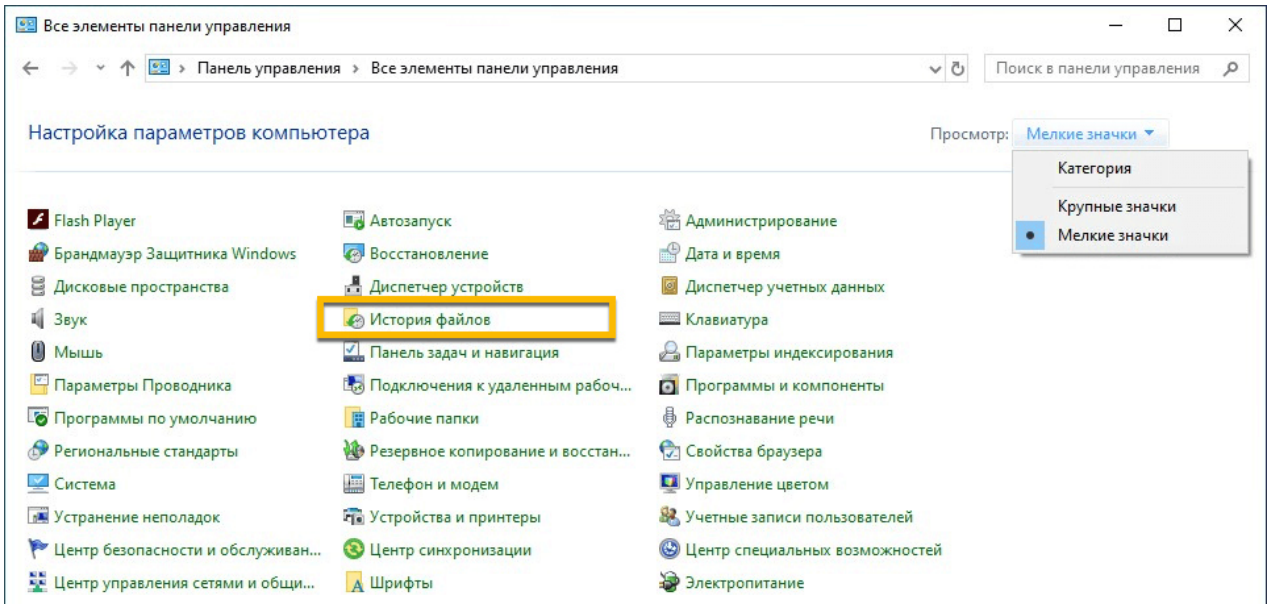

Рисунок 24 – История файлов

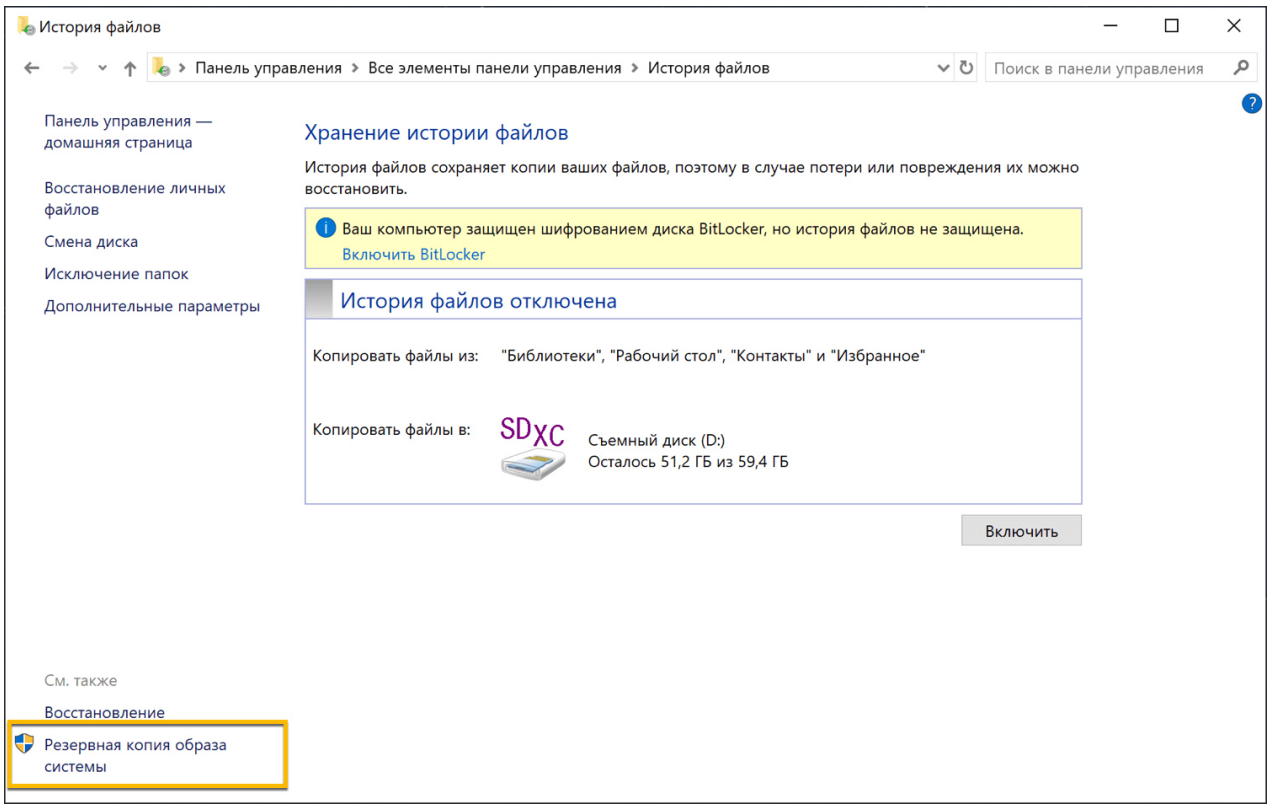

Рисунок 25 – Пункт резервное копирование

Необходимо подключить съемный USB-носитель или SD-карту с достаточным объемом доступного пространства, как показано на рисунке 26.

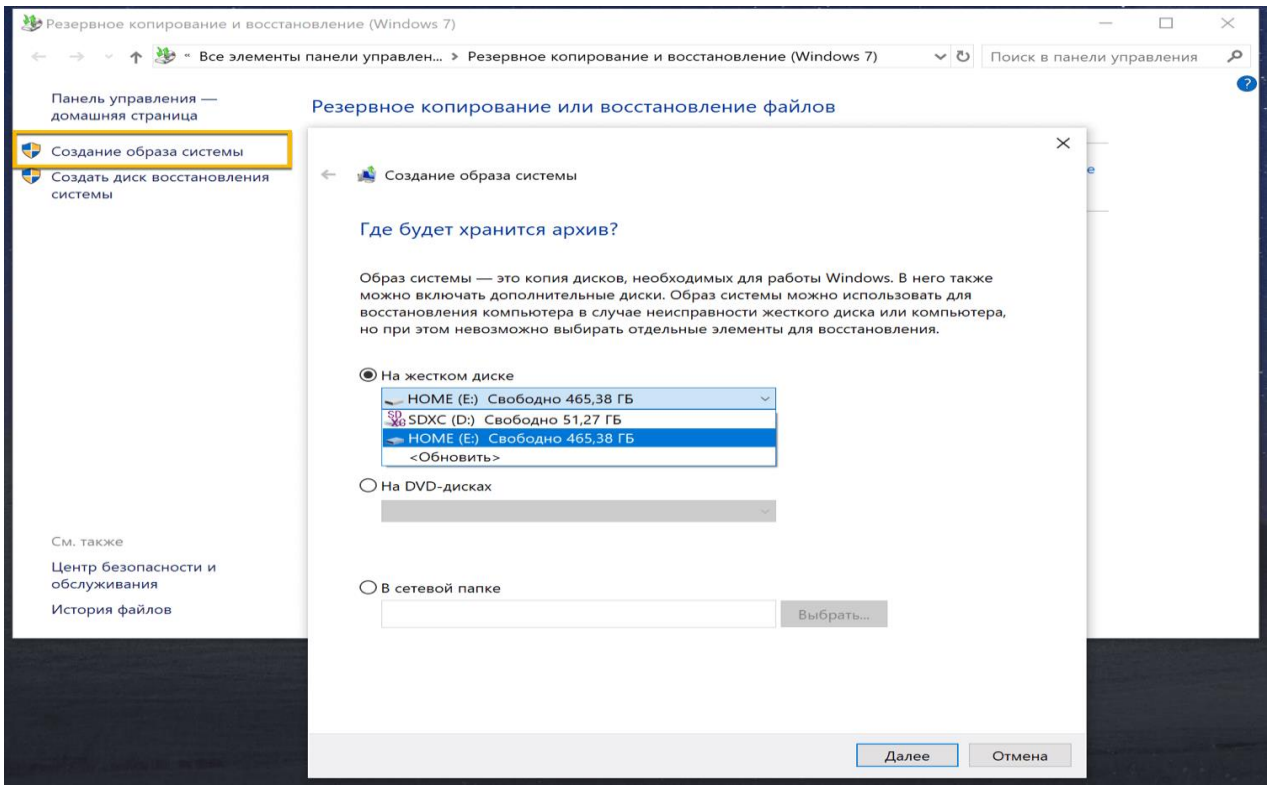

Рисунок 26 – Создание образа системы

Выбор дисков для архивации и процесс резервного копирования представлен на рисунках 27-30.

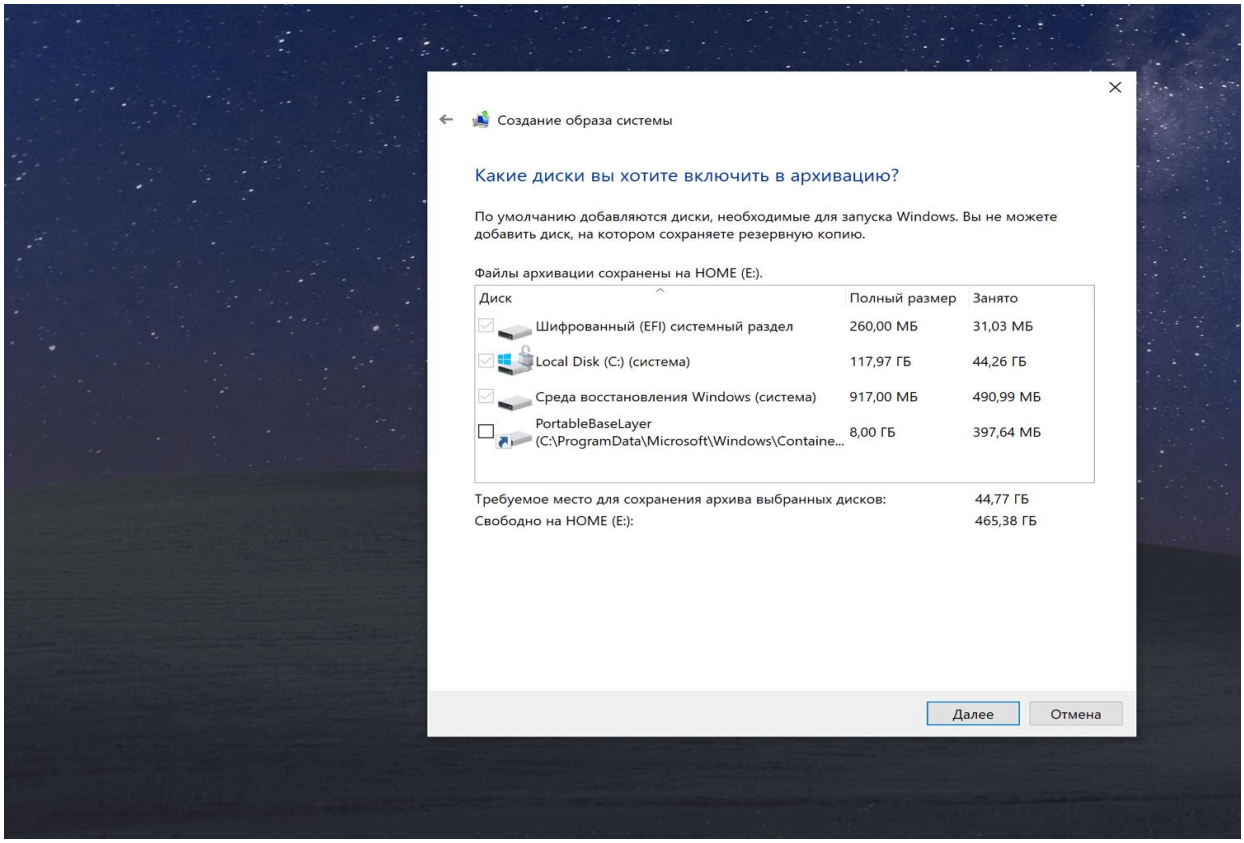

Рисунок 27 – Выбор диска архивации

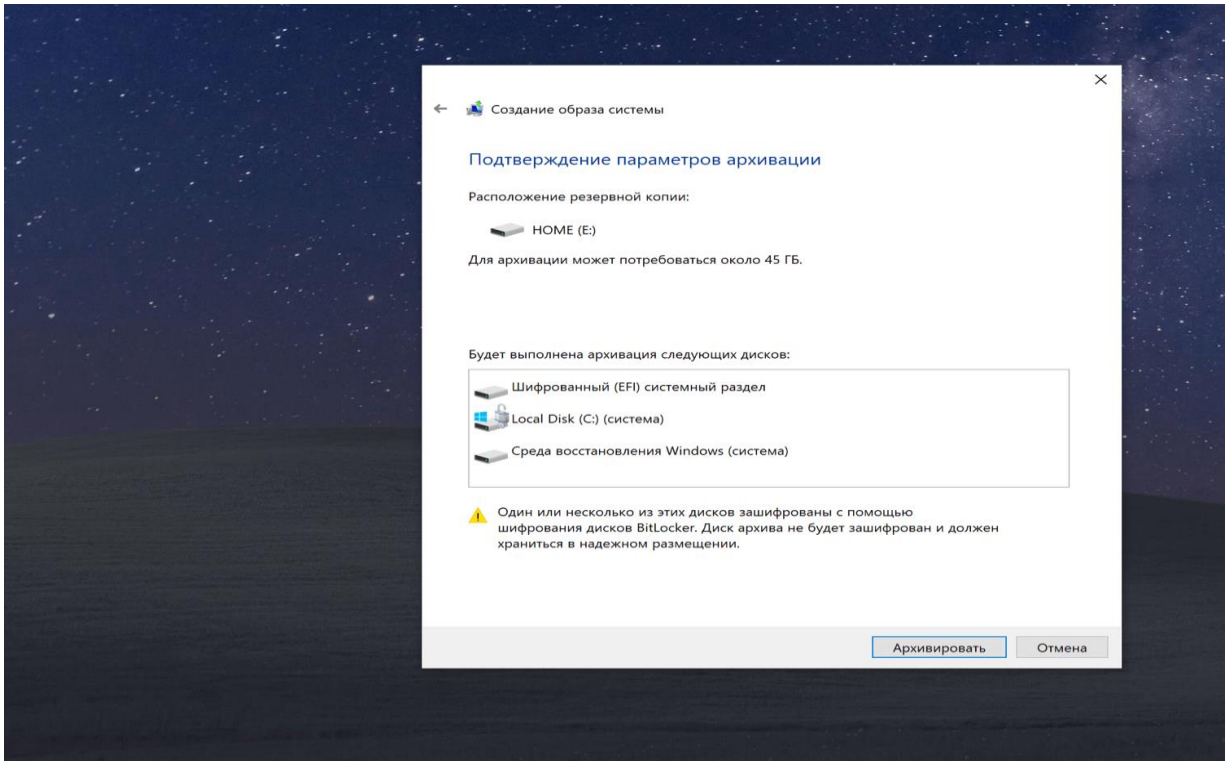

Рисунок 28 – Запуск резервного копирования

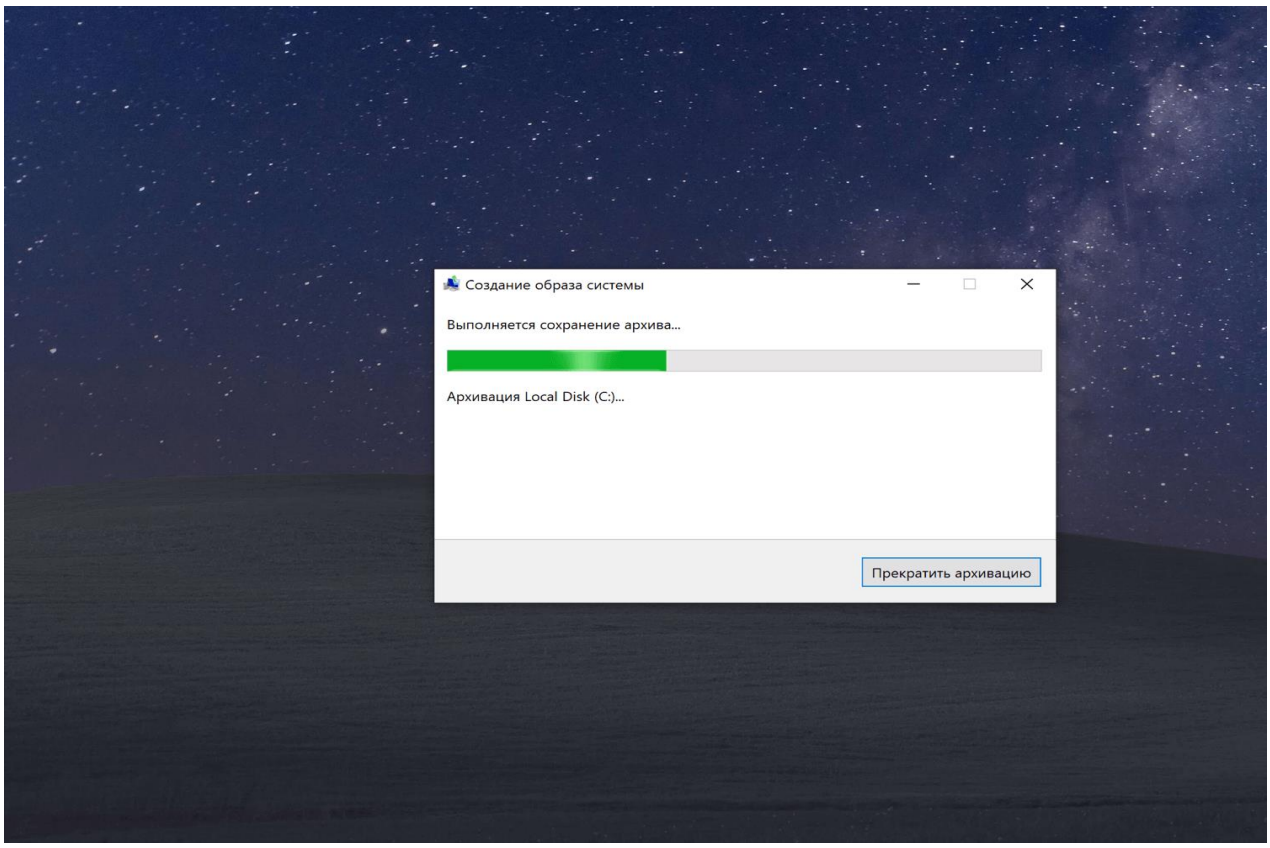

Рисунок 29 – Процесс резервного копирования

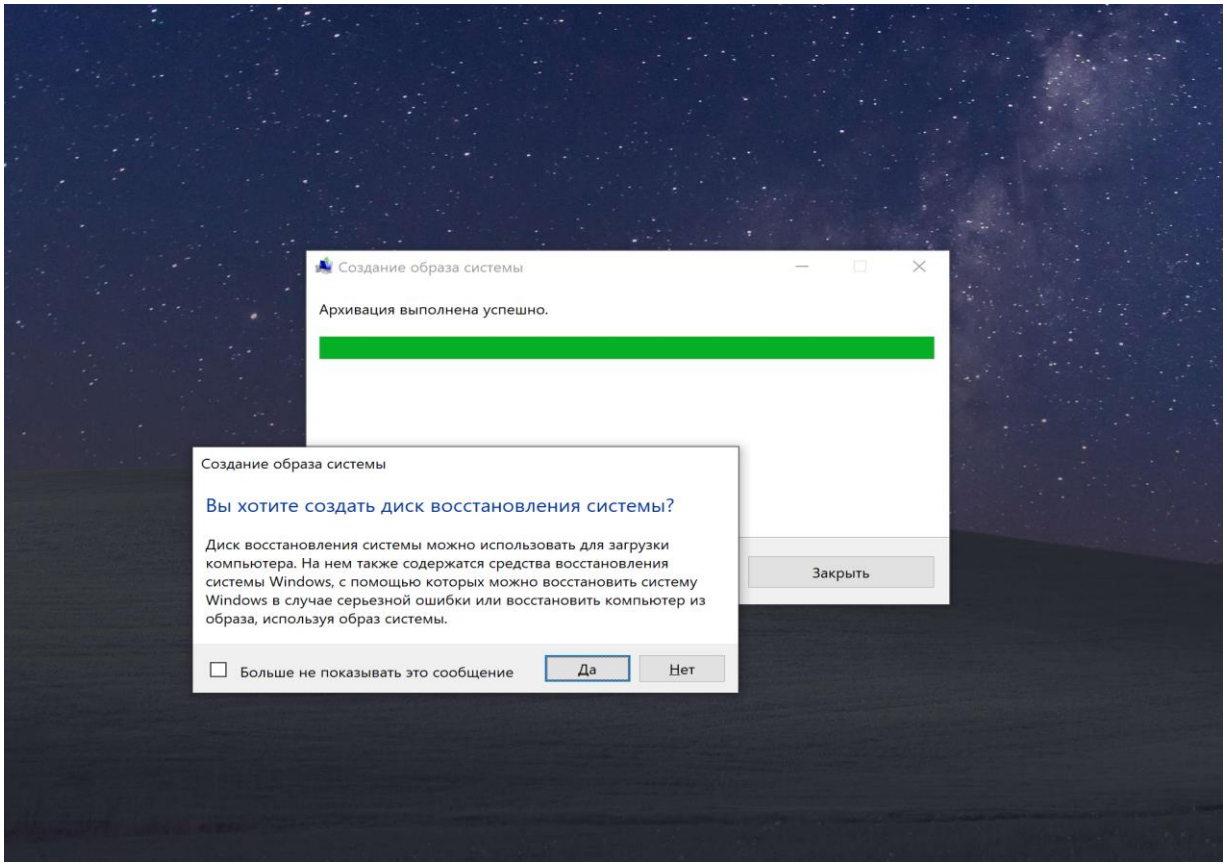

Рисунок 30 – Завершение резервного копирования

Необходимо настроить параметры локальной политики безопасности на ограничение допуска расширения, указать в записи «Политики ограниченного использования программ», кликнув кнопкой мыши на опцию «Создать политику ограниченного использования программ», что показано на рисунке 31-32.

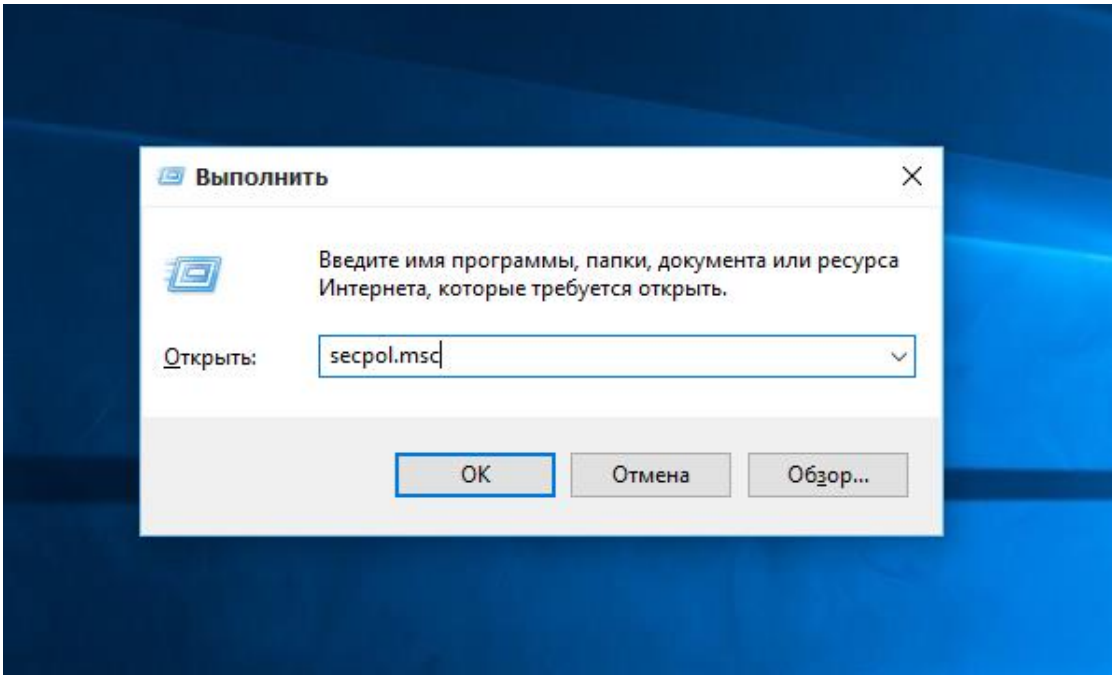

Рисунок 31 – Запуск локальной политики безопасности

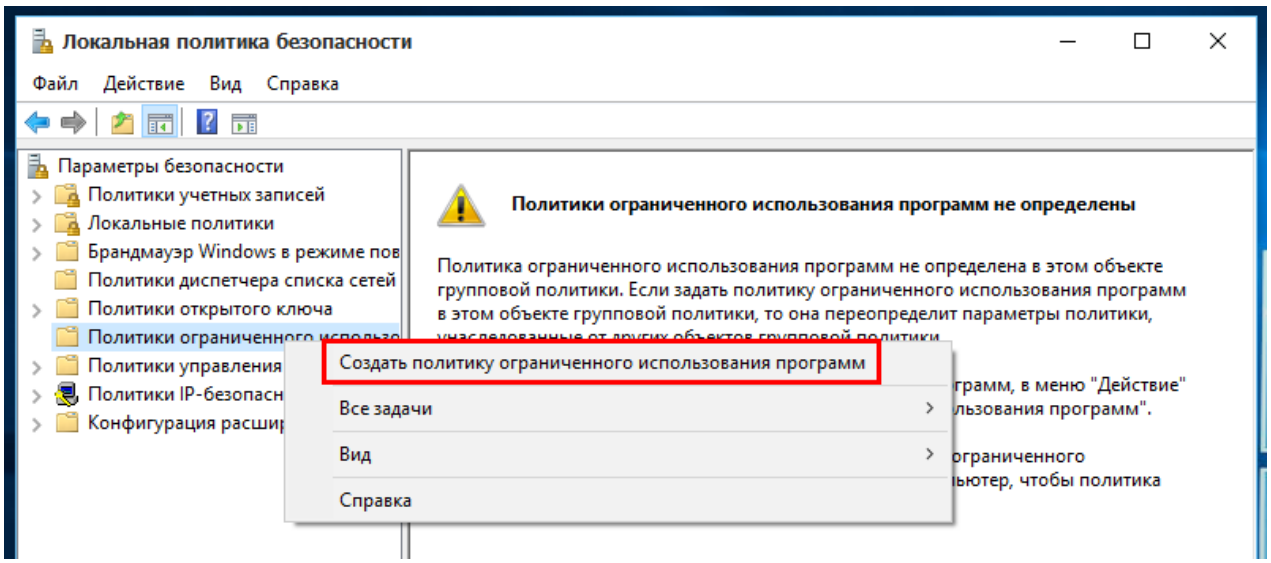

Рисунок 32 – Создание политики

Запустив свойства (рисунок 33), после необходимо указать параметры и выбрать расширение, на которое накладывается ограничение.

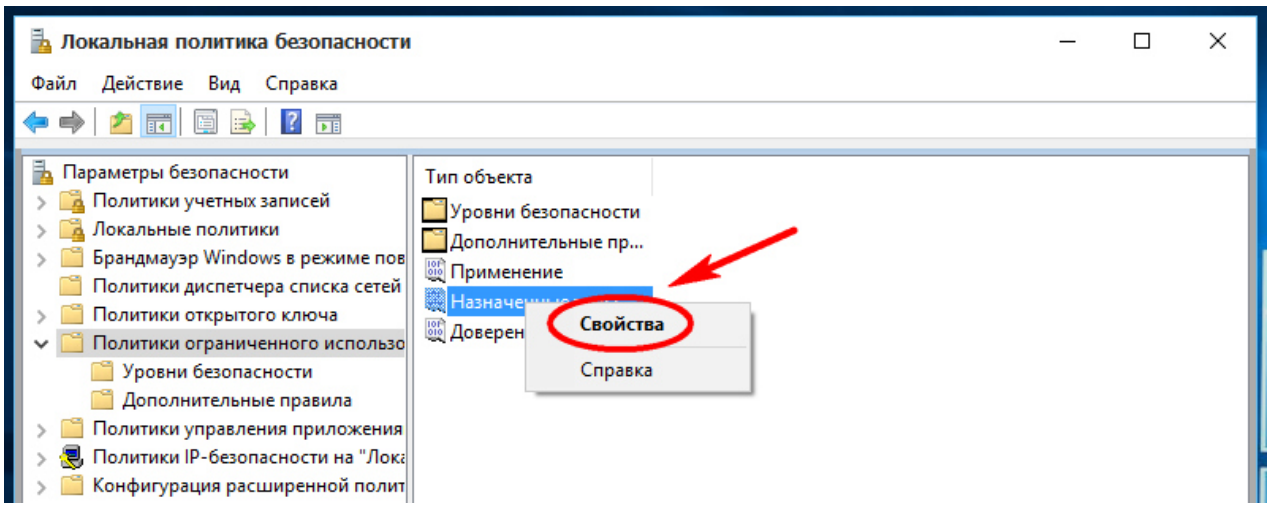

Рисунок 33 – Свойства

Выбрав параметр, применять политику для «всех пользователей, кроме локальных администраторов», рисунок 34.

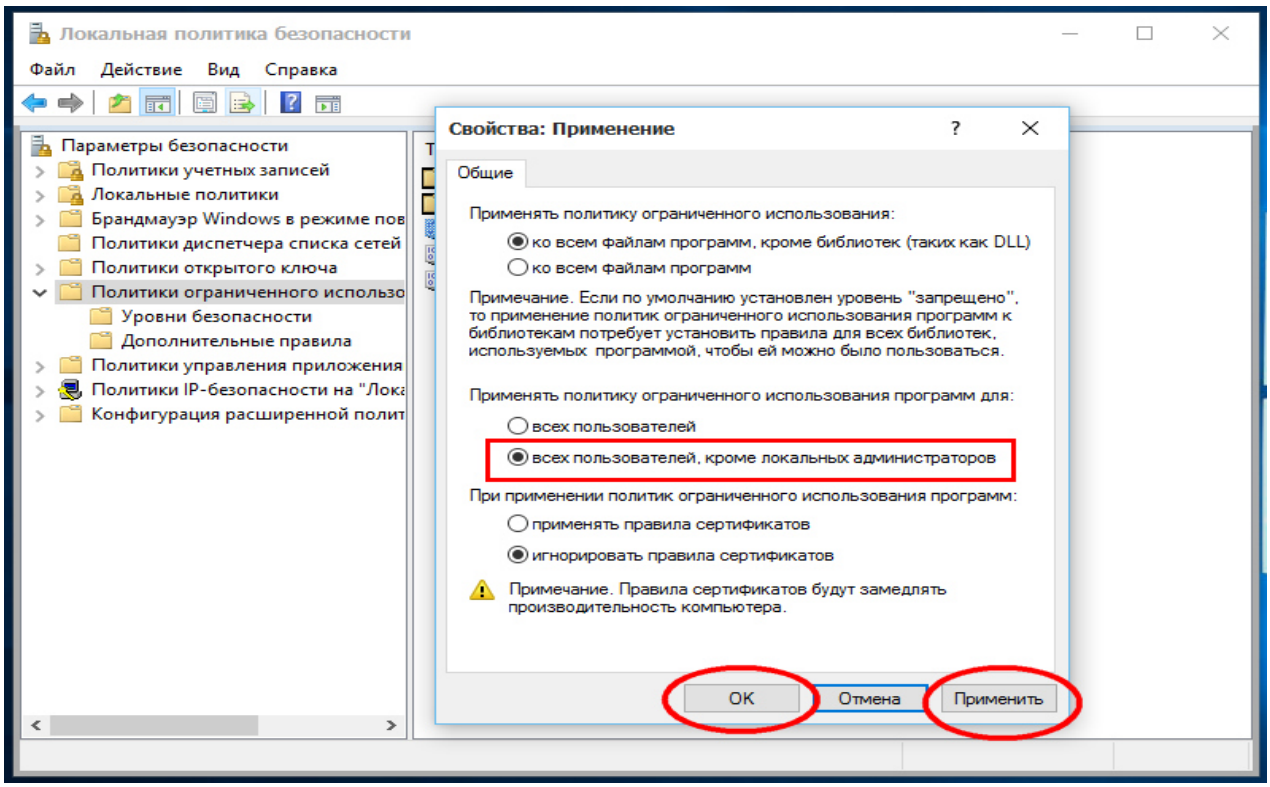

Рисунок 34 – Указание параметра применения

На рисунках 35-36 показан выбор расширений таких как .exe, .js.

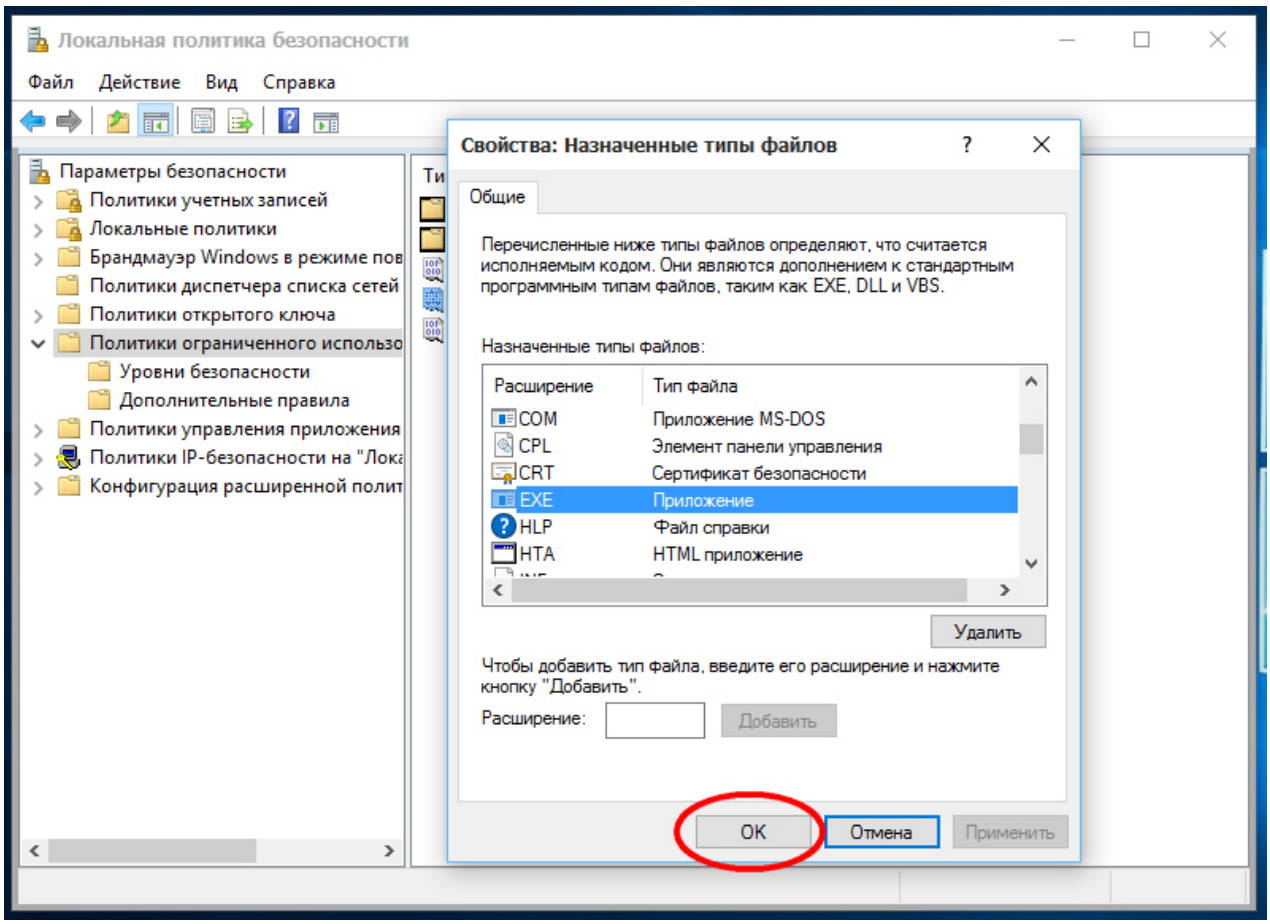

Рисунок 35 – Выбор расширения

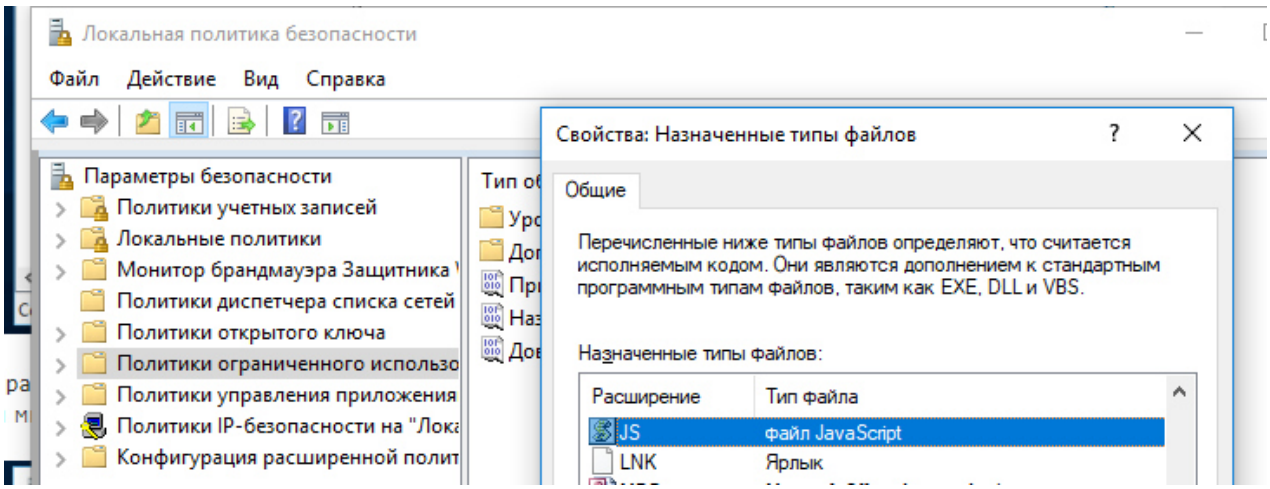

Рисунок 36 – Выбор расширения

Как применяются политики показано рисунке 37.

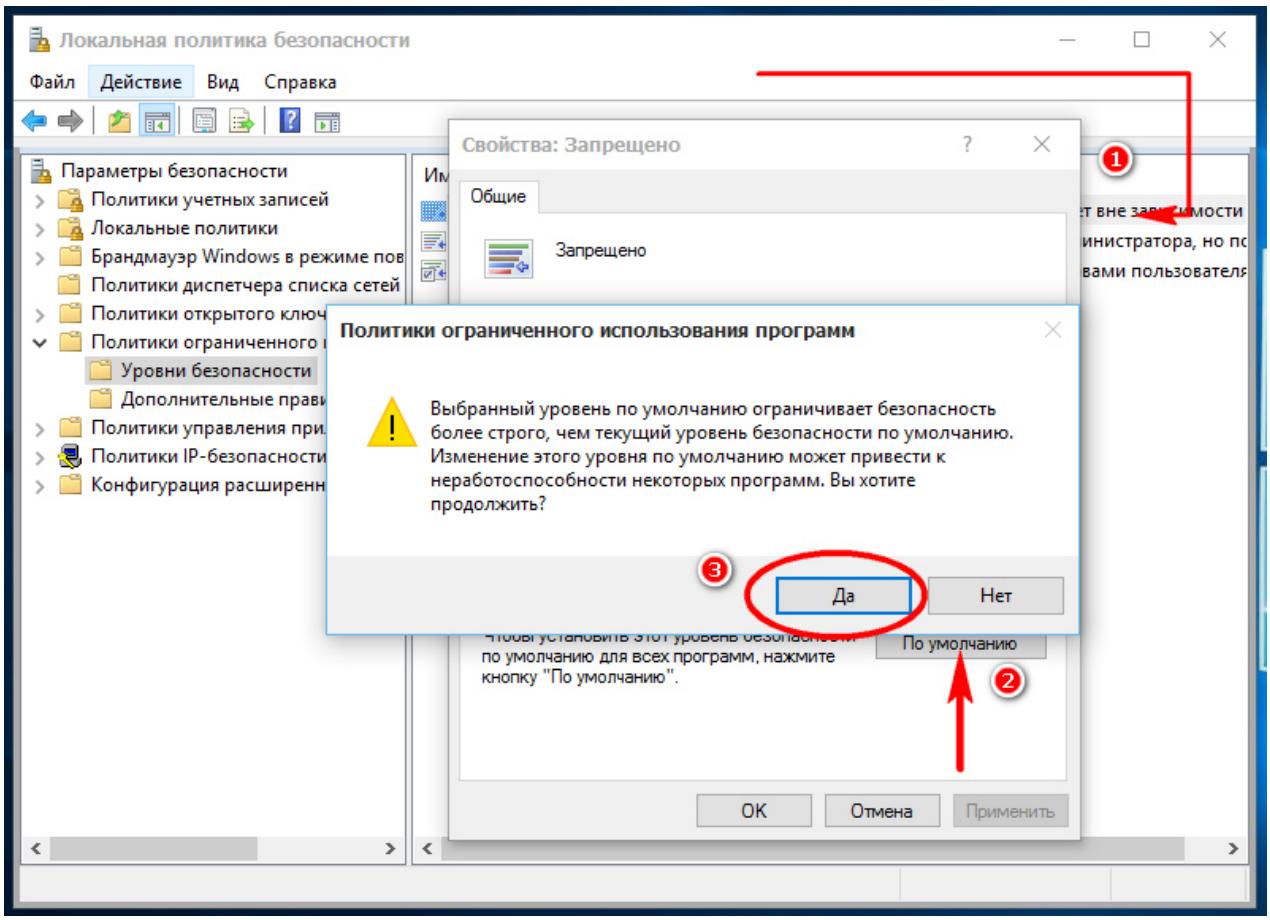

Рисунок 37 – Применение политики

На рисунке 38 показана работа выстроенной политики безопасности, примененной выше, при реакции системы на запуск файла с расширениями .exe или .js.

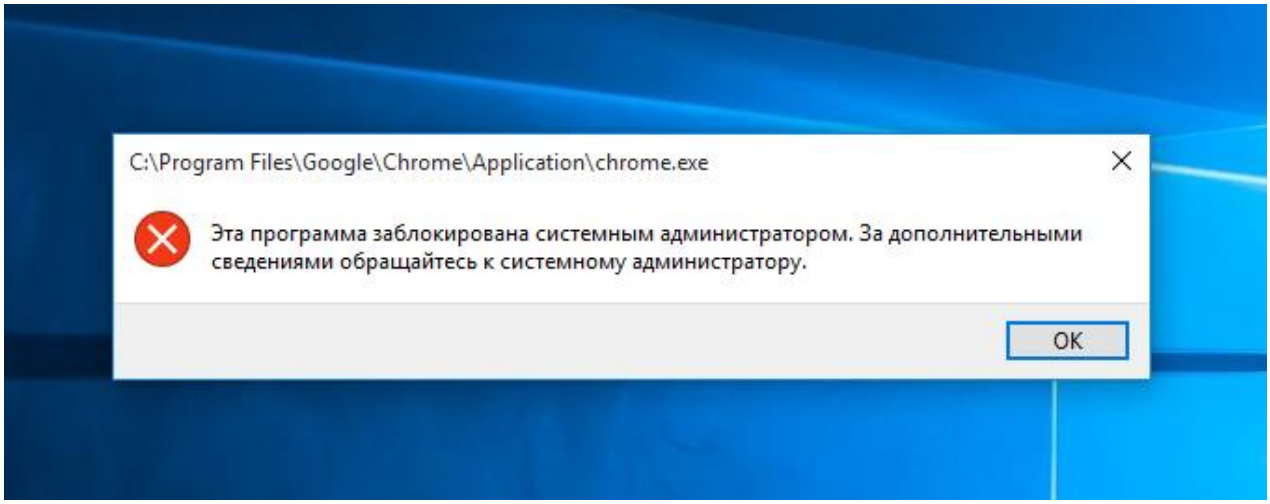

Рисунок 38– Выстроенная политика безопасности в действии

4 Технико-экономическое обоснование

Основной целью данного дипломного проекта является исследование программного кода на уязвимость и последующее воздействие на систему.

Актуальность данной темы заключается в том, что на текущий момент информация является ценным ресурсом, с которым необходимо правильно взаимодействовать, и если допустить вероятность потери данных, то это оценивается в огромном материальном ущербе.

На данный момент существует большое количество вредоносного программного обеспечения. Обратная разработка программного кода необходима для исследования влияния программного обеспечения на систему. В процессе используется достаточное количество специального софта: отладчики, декомпиляторы, распаковщики и прочие необходимые инструменты.

Данный проект будет полагаться лишь на одного специалиста. Он будет выполнять сразу несколько функций: читать код для его обратной разработки, использовать весь необходимый функционал софта, следить за сроками выполнения проекта и т.д.

Технико-экономическое обоснование разработки и внедрения в моей работе содержит следующее:

- определение сложности выполнения обратной разработки кода;
- определение возможной стоимости исследования;
- оценку социально-экономических результатов работы исследования.
- программы.

4.1 Расчет трудоемкости исследования

Приведен список стадий и типов работ, необходимых для определения точной степени трудоемкости обратной разработки кода. Трудоемкость рассчитывается в соответствии с общепризнанными периодами в осуществления расчетов. Модель распределения трудоемкости по стадиям приведена в таблице 1.

Таблица 1- Распределение работ по этапам и оценка их трудоемкости

Этапы разработки ПП	Виды работ	Трудоемкость разработки, чел. х ч.
1 этап	Постановка задач	20
2 этап	Разработка и утверждение технического задания на разработку ПП	25
3 этап	Ознакомление с методиками обратной разработки кода	16
4 этап	Поиск и ознакомление с литературой по теме обратной разработки кода	30

Продолжение таблицы 1

5 этап	Изучение работы необходимого программного обеспечения	18
6 этап	Оформление теоретической части темы дипломного проекта	21
7 этап	Разработка практической части программного проекта	30
8 этап	Реализация проекта	40
9 этап	Оформление отчета и составление выводов о проделанной работе	17
10 этап	Проверка верности исследования	15
11 этап	Итог исследования	16
ИТОГО трудоемкость выполнения программного проекта		248

Трудовой день приравнивается к 8 часам. В таком случае для реализации дипломного проекта потребуется 31 рабочий день.

4.2 Расчет затрат на исследование

Определение затрат на исследование вредоносного программного обеспечения производится на основе имеющейся сметы, которая содержит:

- материальные затраты;
- затраты на оплату дополнительного софта;
- прочие затраты.

Материальные затраты состоят из главных и вспомогательных материалов, энергии, требуемых с целью исследования ПП. Расчет затрат на материальные ресурсы производится по форме, приведенной в таблице 2.

Таблица 2 - Затраты на материальные ресурсы

Наименование материала	Марка	Ед. измерения	Количество	Цена за ед. в тенге	Сумма в тенге
Офисная бумага Zoom	International Paper	Пачка	3	1 200	3 600
Тетрадь А5 (160 листов)	Flip	штук	2	420	840

Ручка	BRUNO VISCONTI	упаковка		500	500

Продолжение таблицы 2

Компьютерная мышь (беспроводная)	Defender Venom GM-640L, Black,	штук	1	7 990	7 990
Итого					12840

Используется ноутбук Lenovo Z710 в котором предусмотрены интегрированная ОС и вспомогательные ПО, в этом случае расходы на лицензионные версии Windows 10 и MS Office, производиться не будут.

Таблица 3 - Затраты на ОС и ПО

Наименование материала	Марка	Ед. измерения	Количество	Цена за ед. в тенге	Сумма в тенге
Ноутбук	Lenovo Z710	Шт.	1	350000	350000
Принтер	Epson	Шт.	1	120000	120000
Модем	Ericsson T073G	Шт.	1	16000	16 000
ОС	Windows 10	Шт.	1	-	-
Доп. софт	Eclipse, подписка на any.run	Шт.	1	15000	15000
Итого					513930

Сумма общих расходов на ПО и исследование определяется в соответствии с формулой:

$$Зм = \sum Pi \times Цi, \quad (1)$$

где Pi - расход i-го вида материального ресурса, натуральные единицы;
Цi - цена за единицу i-го вида материального ресурса, тг;
i - вид материального ресурса;
n - количество видов материальных ресурсов.

Зм =3600+840+500+7990+350000+120000+16000+15000= 513 930 (тг)

Материальные затраты на разработку программного проекта составили:

513 840 тенге.

4.3 Расчет затрат на электроэнергию

В ходе выполнения работы по разработке программного проекта применяется электрооборудование, соответственно следует вычислить расходы на электроэнергию.

Время работы оборудования для достижения цели по исследованию программного кода равняется 248 часам (см. таблица таблицы 1. Для принтера период деятельности равняется 3 часа, так как постоянная потребность в его использовании отсутствует.

$$Э = З_{эл.эн.обор} + З_{доп.нуж}, \quad (2)$$

где $З_{эл.эн.обор}$ – затраты на электроэнергию оборудования;

$З_{доп.нуж.}$ – затраты электроэнергии на дополнительные нужды.

Расходы электроэнергии на оборудование определяются в соответствие с формулой:

$$З_{эл.эн.обор} = \sum W \times К_{исп} \times S \times T, \quad (3)$$

где W – потребляемая мощность, Вт;

$К_{исп}$ – коэффициент использования ($К_{исп}$ = 0,7..0,9);

T – время работы;

S – тариф (1кВт/ч = 23,85тг).

Пример расчета затрат на электроэнергию оборудования

Зэ = 0,7*0,8*248*23,85 = 3312.3

Итоги расчета расходов на электроэнергию представлены в таблице 4.

Таблица 4 - Затраты на электроэнергию

Наименование приборов	Паспортная мощность, кВт	Коэффициент мощности	Время работы оборудования, ч	Цена ЭЭ тг/ кВтч	Сумма, тг
Ноутбук	0,7	0,8	248	23,85	3312,3
Модем	0,09	0,9	200	23,85	386,3
Принтер	0,4	0,9	3	23,85	25,7
Кондиционер	0,9	0,9	238	23,85	4597,8
Освещение	0,4	0,7	248	23,85	1656,14
Итого					9978,24

$$З_{эл.эн.обор} = 3312,3+386,3+25,7+4597,8+1656,14= 9978,24(тенге)$$

Расходы на дополнительные потребности принимаются согласно укрупненному показателю в объеме 5% от расходов на оборудование:

$$З_{доп.нуж} = 5\% \times З_{эл.эн.обор}, \quad (4)$$

Затраты на дополнительные потребности определяется в соответствие с формулой (4.4):

$$З_{доп.нуж} = 0,05 \times 9978.24 = 498,9 \text{ (тенге)}$$

Таким образом итоговые расходы в электроэнергию составляют:

$$Э = 9978,24+498,9 = 10477,2 \text{(тенге)}$$

4.4 Амортизация основных фондов и прочие затраты

Годовые амортизационные нормы ОФ берутся в соответствии с налоговым кодексом РК.

Амортизация основных фондов согласно формуле (5):

$$Аг = \frac{Соб \cdot На}{100}, \quad (5)$$

где $С_{об}$ – стоимость оборудования;

$Н_А$ – норма амортизации (норма амортизация = 25);

По формуле 0, производится рассчет необходимой суммы для амортизационных отчислений за год для ноутбука:

$$Аг = \frac{350000 \cdot 25}{100} = 87500 \text{ тг}$$

Сумма амортизации за период разработки:

$$Аг = \frac{87500 \cdot 31}{365} = 7431,5 \text{ тг}$$

Произведем расчёт, суммы амортизации для остального оборудования

Таблица 5- Амортизация основных фондов (ОФ)

Наименование оборудования и ПО	Стоимость оборудования и ПО, тг	Годовая норма амортизации, %	Сумма амортизации за год, тг	Сумма амортизации за время разработки, тг
Ноутбук	35000	25	87500	7431,5
Принтер	120000	25	30000	2547,94
Модем	16 000	25	4000	339,72

Доп. софт	15000	25	3750	318,49
ИТОГО			125250	10637,65

Смета расходов на исследование программного кода.

Основываясь на произведенные расчёты и полученные сведения, согласно единичным заметкам, формируется смета на исследование программного кода, на рисунке 39 приведена диаграмма рабочих затрат.

Таблица 6– Смета затрат на исследование программного кода

Статьи затрат	Сумма, тг
Затраты на оборудование	498930
Затраты на программное обеспечение	15000
Затраты на электроэнергию	10477,2
Амортизация основных фондов	10637,65
Доп. софт	15000
Итого по смете	550044,85

Рисунок 39 – Диаграмма рабочих затрат

Расчет договорной цены ($Ц_Д$) производится по формуле:

$$Ц_д = З_{нир} \cdot \left(1 + \frac{Р}{100}\right) \quad (6)$$

$Ц_д$= 550044.85 * (1+ 20/100) = 660053.8

Значение прибыли составило 110008.9

Прибыль = 5500044.85 * 0.2 = 110008.9

Цена реализации с учетом НДС составила 739260.3

$Ц_р$=660053.8 + (660053.8 * 0,12) = 739260.3

Вывод

В данной главе дипломного проекта были произведены вычисления экономических расходов для приобретения необходимого программного обеспечения, оборудования требуемого для произведения исследования программного кода вредоносных программ. Были рассчитаны все расходы на: приобретение оборудования, расчёт трудоёмкости исследования, электроэнергию, амортизационные отчисления и т.д.

В смете, составленной мной после произведения расчётов окончательным показателем затрат, составило: 550044,85 тенге. В показатели входят данные по: затратам на оборудование, затратам на ПО, затратам на электроэнергию, затратам на амортизацию ОФ и прочим затратам. Прибыль составила: 110008.9 тенге. Цена реализации с учетом НДС составила 739260.3 тенге.

5 Безопасность жизнедеятельности

Основной целью данного дипломного проекта является исследование программного кода на уязвимость и последующее воздействие на систему. Актуальность данной темы заключается в том, что на текущий момент информация является ценным ресурсом, с которым необходимо правильно взаимодействовать, и если допустить вероятность потери данных, то это оценивается в огромном материальном ущербе.

Данное исследование может использоваться различными компаниями или просто пользователями, которым приходится сталкиваться с фишинговыми рассылками. Благодаря этому исследованию подверженные заражению системы могут быть спасены, поскольку будет известно какими методами можно вылечить систему, где хранится главный зараженный документ, какие профилактические меры стоит проводить, помимо общеизвестных и достаточно много других факторов, которые помогут сэкономить компании большие средства. Если вы не соблюдаете меры безопасности пользования интернет ресурсом, в частности почтой, то велик риск заражения именно тем вредоносным программным обеспечением, о котором и говорится в моем дипломном проекте.

5.1 Анализ условий труда

При исследовании кода программного обеспечения, работник вынужден долгое время взаимодействовать с компьютерной техникой. Рабочая зона – это та зона временного либо постоянного присутствия работника. Из за того что работник должен проводить длительное время на стуле в сидячем положении должны быть предусмотрены меры максимального удобства, которые позволят работать уютно и без вредного воздействия. Эти меры должны включать в себя: компьютерное оборудование и мебель необходимо размещать оптимальным образом, достаточное рабочее пространство, которые позволит работнику проделывать все необходимые действия и перемещения, работник должен получать необходимое количество световых лучей, чтобы максимально снизить нагрузку на зрение, на рабочем месте должна соблюдаться комфортная комнатная температура при которой работник сможет чувствовать себя комфортно.

Существует несколько типов освещения, естественное и искусственное.

Естественное освещение – освещение, которое проникает через световые проемы, и является дневным светом. Этот тип освещения меняется в связи с природными условиями, временем суток, временем года.

Искусственное освещение – освещение, в котором не участвует естественное освещение, и может использоваться в тёмное время суток и тогда, когда естественного освещения не хватает.

Использование искусственного и естественного освещения одновременно, называется комбинированным освещением.

Правильность выбора подсветки и освещения в виде светильников и ламп, а также корректное расположение обеспечит не привыкающую значения 40 кд/м^2 отраженность бликов на рабочей станции и рабочей поверхности.

Для искусственного освящения должны применяться люминесцентные лампы белого света. В производственной среде и административно-общественных помещениях можно использовать металл галогенные лампы мощностью до 250 Вт.

5.2 Характеристики рабочего помещения

Рабочее помещение, в котором работник проводит свое исследование рассчитан на одно рабочее место. Располагается на 4 этаже жилого здания. Визуализированная модель помещения представлена на Рисунке 42.

Характеристики помещения: длина L = 5 метра, ширина B = 4 метров, высота H = 3 метра. Помещение было построено и оборудовано согласно санитарным требованиям от 01.12.2011 года, т.е. площадь одного рабочего места 4,5 метра в квадрате, а монитор должен находиться на расстоянии 60см от глаз. Рабочее пространство работника удовлетворяет требованиям и составляет 20м^2.

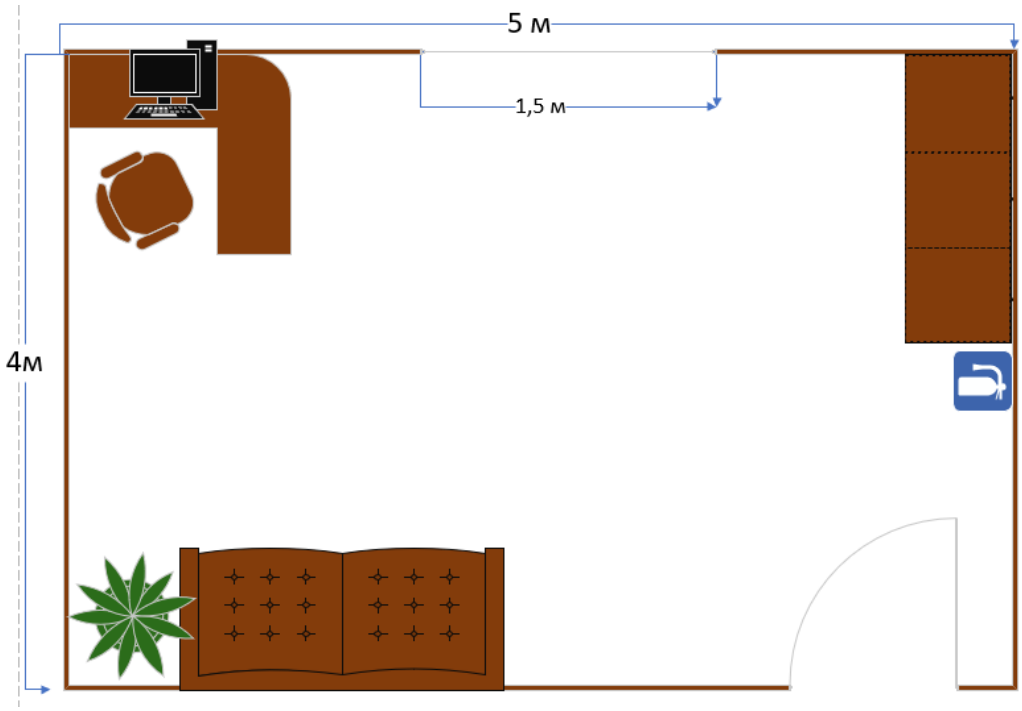

Рисунок 42 – Рабочее помещение

5.3 Используемое оборудование и его характеристики

Ноутбук Lenovo IdeaPad Z710. Технические характеристики устройства:

- Intel(R) Core i7 4710MQ (CPU 2.5 GHz);
- Intel HD 4600;
- ОЗУ 8 ГБ;

- HDD 1 ТБ;
- электропитание: 220-250В, 50Гц, 400 Вт;
- габариты(мм): 270 - 414 – 32,5.

Модем: 4-х портовый с коммутатором 10/100 Мбит/с.

Стул: высота 0,6 м.

Стол: высота – 0,8 м, длина – 4 м, ширина – 1 м.

5. 4 Расчет естественного освещения

Один из главных показателей, который должен находить на заданном уровне, это освещение. Качество освещения важно для создания удобства при работе. Хороший уровень освещения необходим для комфортной работы и исключения зрительного напряжения, которое в следствии может привести к ухудшению здоровья и сказаться на качестве работы. Согласно нормам освещенности (СНиП 11-4-79) и отраслевым нормам, работа инженера относится к четвертому разряду зрительной работы. Важно проводить расчеты по освещению, они необходимы для определения площади световых проёмов естественного освещения и характеристики искусственного освещения. Формула (7) для расчета площади световых проемов при естественном освещении:

(7)

$$S = \frac{(S_n * e_n * K_n * h_0 * K_{зо})}{(100 * t_0 * r_1)}$$

где,

S_n – площадь помещения, м^2;

e_n – нормированное значение КЕО, %;

K_n – коэффициент запаса;

h_0 – световая характеристика окон (6,5 - 29);

$K_{зо}$ – коэффициент затемнения окон зданиями, стоящими напротив (1,0-1,7);

r_1 – коэффициент повышение КЕО за счет отраженного света от поверхности помещения (1,05 - 1,7);

t_0 - общий коэффициент светопропускания равен от 0,1-0,8.

Полагаясь на данные характеристики, длина помещения равна 5 метров, а ширина равно 4 метра, можно найти площадь пола по следующей формуле:

(8)

$$S_n = L * B$$

$$S_n = 5 * 4 = 20 \text{ м}^2$$

Для данного рабочего пространства площадь световых проемов естественного бокового освещения определяется формулой (7), необходимы следующие значения:

Где,

$e_n = 1{,}5$ %;

$К_n = 1{,}5$;

$h_0 = 29$;

$К_{зо} = 1{,}1$;

$r_1 = 1{,}3$;

$t_0 = 0{,}7$.

Теперь необходимо подставить значение данных коэффициентов в формулу (0,1) и вычислить площадь световых проемов:

$$S = \frac{20 * 1{,}5 * 1{,}5 * 29 * 1{,}1}{100 * 0{,}7 * 1{,}3} = 15{,}77 \text{ м}^2$$

Рассчитав площадь оконного пространства, значение вышло 15.77м^2, из чего следует вывод что необходимо искусственное освещение так как окна площадью 1.5м^2 недостаточно для создания комфортных условий освещения.

5.5 Расчет искусственного освещения

Основываясь на норму СНиП 11-4-79 для четвертого класса зрительных работ освещенность помещения должно быть не менее 200 Лк. Номинальная освещенность рабочего места определяется формулой:

(9)

$$E = \frac{Ф_{св} * n * N * 1}{s * К_з * Z}$$

Где,

$Ф_{св}$ – световой поток от ламп, Лк;

N – количество светильников;

$К_з$ – коэффициент, учитывающий запыленность светильников;

n – коэффициент использования светильников;

s – площадь помещения, м^2;

z – коэффициент неравномерности освещения.

Основываясь на норму СНиП 11-4-79 для типа ламп, который будут использоваться $К_з = 1{,}4 : 1{,}5$ при нормальной эксплуатации светильников; z = 1,1:1,2 при оптимальном их размещении. В данном случае коэффициент n полностью зависит от светильников и их типа, коэффициенты отражения светового потока от потолка - p_2, от пола p_3, от стен p_1, зависят от размера помещения, учитывающих величиной I, где I это индекс помещения.

(10)

$$I = \frac{(A * B)}{h_c * (A + B)}$$

где A, B - параметры помещения, м;

h_c - высота светильников над рабочей поверхностью.

Расчёт высоты светильников над рабочей поверхностью выводится по формуле:

(11)

$$h_c = H_{помещения} - H_{свеса} - H_{р.п.}$$

где $H_{свеса}$ = 0,4 - высота свеса ламп, м;

$H_{р.п.}$ = 0,8 - расстояние рабочей поверхности над полом, м;

$H_{помещения} = 3$ - высота помещения, м.

Основываясь на формулу (11) определяется высота светильников над рабочей поверхностью:

$$h_{расч} = 3 - 0{,}4 - 0{,}8 = 1{,}8 \text{ м}$$

Зная что параметры помещения равны 4м × 5м и высота светильников над рабочей поверхностью $h_c = 1{,}8$м, по формуле (10):

$$I = \frac{(4 * 5)}{2 * (4 + 5)} = 1{,}1$$

Основываясь на таблицу (7) определяется коэффициент использования светового потока n, учитывая, что коэффициенты p_1 = 30%, $p_2 = 50\%$, $p_3 = 10\%$.

Таблица 6 - Значения коэффициента использования светового потока

Коэффициент I	0,5	1	2	3	4
Коэффициент использования светового X потока, h	0,22	0,36	0,48	0,54	0,59

Коэффициент n = 0,3 для рабочего места. От лампы типа ДРЛ-80 световой поток равняется 3800 Лк, в совокупности от 3 ламп световой поток будет равняться 11400 Лк. Основываясь на все вычисления и данные можно определить номинальную освещенность рабочего места по формуле (9).

$$E = \frac{11400 \cdot 0{,}3 \cdot 3}{20 \cdot 1{,}4 \cdot 1{,}2} = 305{,}3 (Лк)$$

Значение, полученное в ходе расчётов, соответствует нормальным условиям освещенности и создается комфортную для работы обстановку.

5.6 Расчет воздухообмена в рабочем помещении

При работе важной характеристикой является температура и вентиляция. Условия труда в которых используется оборудование для регулирования температуры, такие как вентиляторы, кондиционеры или естественная вентиляция, представляемая от окон или дверей, необходимо рассчитывать по формуле:

(12)

$$Q = \frac{g}{X - X_n}$$

где Q - потребный воздухообмен, $м^3$/ч;

g - количество вредных веществ, л/ч;

X – предельно допустимая концентрация вредных веществ в помещения, $л/м^3$;

X_n – предельно допустимая концентрация вредных веществ в наружном воздухе, $л/м^3$.

Кратность воздухообмена n, показывает количества раз в течении часа воздух обязан смениться.

(13)

$$n = \frac{Q}{Q_{пот}}$$

где $Q_{пот}$ это объем помещения;

Объем помещения $Q_{пот}$ определяется формулой:

(14)

$$Q_{пот} = L * H * B$$

$$Q_{пот} = 5 * 4 * 3 = 60 м^3$$

Количество углекислого газа, который выделяется человеком при работе в расслабленном состоянии равен 23 л/ч, в помещении предельная концентрация не должна превышать 1 $л/м^3$, предельно допустимая концентрация в наружном воздухе 0,5 $л/м^3$. Потребный воздухообмен определяется по формуле (12):

$$Q = \frac{23}{1 - 0{,}5} = 46 \left(\frac{м^3}{ч}\right)$$

Формулой (13) определяет кратность воздухообмена n:

$$n = \frac{46}{60} = 0{,}7$$

Воздухообмен для удаления избыточного тепла рассчитывается по формуле:

(15)

$$Q_{изб} = \frac{L_{изб}}{G_B * C_B * d_t}$$

где $L_{изб}$ - избыточное тепло, ккал/ч;
G_B = 1,207 кг/м^3 - удельная масса приточного воздуха;
C_B = 0,25 ккал/кг·°С - теплоемкость воздуха;
d_t - разность температур удаленного и приточного воздуха;
d_t зависит от тепло напряжённости воздуха - L_H. Если L_H больше 20 ккал/ч, то d_t = 8°С. Если L_H меньше 20 ккал/ч, то d_t = 6°С.
Тепло напряжённости воздуха определяется формулой:

(16)

$$L_H = \frac{L_{изб}}{Q_{пот}}$$

Количество избыточного тепла определяется формулой:

(17)

$$L_{изб} = L_{об} + L_{ос} + L_л + L_р + L_{отд}$$

где $L_{об}$ - тепло от оборудования, ккал/ч;
$L_{ос}$ - тепло от системы освещения, ккал/ч;
$L_л$ - тепло, выделяемое людьми, ккал/ч;
$L_р$ - тепло от солнечной радиации, ккал/ч;
$L_{отд}$ – теплоотдача естественным путем, ккал/ч.
Тепло от оборудования определяется формулой:

(18)

$$L_{об} = 860 * P_{об} * f$$

где $P_{об}$ - номинальная мощность оборудования, Вт;
f - коэффициент передачи, f = 0,25.
Тепло от системы освещения определяется формулой:

(19)

$$L_{ос} = 860 * P_{ос} * a * b * cof(f)$$

где $P_{ос}$ - номинальная мощность освещения, кВт;
а – коэффициент перевода электрической энергии в световую, 0,46;

b – коэффициент одновременной работы ламп, b = 1;
cos(f) – коэффициент мощности, cos(f) = 0,3.
Тепло выделяемое людьми определяется формулой:

$$L_л = n * g \quad (20)$$

где n - количество человек;
g - тепловыделение одного человека, g=50 (ккал/ч).
Тепло от солнца для одного окна определяется формулой:

$$L_р = F * D_{ос} \quad (21)$$

где F - площадь окна, м^2;
$D_{ос}$ - солнечная радиация, $D_{ос}$ = 65 (ккал/ч).
Тепло излучающее системой, оборудованием, людьми и солнцем для помещения определяется формулами (17-20):

$$L_{об} = 860 \cdot 1 \cdot 0{,}25 = 215 \text{ (ккал/ч)}$$
$$L_{ос} = 860 \cdot (0{,}04 \cdot 3) \cdot 0.46 \cdot 1 \cdot 0.3 = 14{,}24 \text{ (ккал/ч)},$$

$$L_л = 0{,}7 \cdot 50 = 35 \text{ (ккал/ч)},$$

$$L_р = 1{,}5 \cdot 65 = 97{,}5 \text{ (ккал/ч)}.$$

Естественную теплоотдачу приравнивается к $L_р$ в холодное время года и в погоду равной нулю в теплое время года, определяю по формуле (16) можно узнать количество избыточного тепла:

$$L_{изб} = 215+14{,}24+35+97{,}5+0 = 361{,}74 \text{ (ккал/ч)}.$$

Воздушную тепло напряжённость определяется формулой (15):

$$L_Н = \frac{361{,}74}{60} = 6{,}029 \text{ (ккал)}$$

Поскольку тепло напряжённость воздуха меньше 20, $d_t = 6°C$. Используется формула (14) для того, чтобы произвести расчет значения необходимого для воздухообмена и удаления избыточного тепла:

$$Q_{изб} = \frac{361{,}74}{(1{,}206 \cdot 0{,}24 \cdot 6)} = 208{,}3 \text{ (м3/ч)}$$

Исходя из полученных результатов, для удаления лишнего тепла и очистки воздуха нужно использовать вентиляционную систему, которая способна обеспечить требуемую подачу воздуха $Q_{изб} = 208{,}3\ (м3/ч)$. В данном случае подойдет кондиционер Hisense. Данный кондиционер способен обеспечить подачу воздуха до 600 $м^3$ /ч.

Рисунок 43 – Кондиционер Hisense

Технические характеристики:
- мощность (охлаждение): 3.75 квт;
- мощность (обогрев): 3.8 квт;
- потребляемая мощность при охлаждении: 1140 вт;
- потребляемая мощность при обогреве: 1030 вт;
- обслуживаемая площадь: 30 м2;
- уровень шума внутреннего блока: 23-39 дб;
- уровень шума внешнего блока: 55 дб;
- цвет: белый.

Характеристики подключения:
- вентиляция: 600 м3/час;
- класс энергоэффективности при охлаждение/обогреве: a++/a+;
- напряжение/частота: 220 в / 50 гц;
- энергопотребление в режиме ожидания не более 1 вт.

Вывод

В это разделе моего дипломного проекта я рассмотрела и рассчитала световые и воздушные показатели для условий труда. Эти показатели одни из важнейших при организации работы, должны всегда соответствовать рамкам стандартов и нормы, поскольку это будет способствовать созданию благоприятных условий для работника и не будет мешать работе затормаживая ее. Основываясь на расчёты, могу сказать, что, для того чтобы осветить комнату площадью 20 $м^2$ абсолютно не хватает естественного освещения, предоставляемого с окна размером 1.5 метра в длину и 1.5 в ширину. Для удобной работы необходимо комбинированное освещение, в которое включено как естественное, так и искусственное освещение. Полагаясь на расчёты полученные в ходе выполнения этого раздела должно использоваться 3 лампы в моем случае это ДРЛ-80 3800 Лк. Если необходимые условия буду соблюдены, то работник может проводить все необходимые работы и исследование в ночное время.

Следующим важным моментом является расчёт воздухообмена и вентиляции. Полагаясь на результаты расчётов, могу сказать, что для создания хороших условий труда необходим один кондиционер с подачей воздуха не менее 208,3 $м^3$ /ч, в моем случае используется кондиционер Hisense с подачей воздуха до 600 $м^3$ /ч..

Заключение

В данном дипломном проекте была произведена обратная разработка кода вируса-шифровальщика. Каждый вирус-шифровальщик имеет свой алгоритм шифрования, а бывает даже несколько. Недостаточно знать лишь алгоритм, чтобы подобрать ключ дешифрования. Обнаружив исходный код вируса, можно лишь выяснить насколько глубоко было проникновение, и какие файлы системного уровня были подвержены инфицированию, какие данные были удалены, а какие изменены. Вирус-шифровальщик, который был исследован в данном дипломном проекте, содержался в архиве, и передавался по почте (фишинговой рассылкой). Пользователь, скачивая вложение в виде вируса, становился лишь потенциальной жертвой, но из-за малой осведомленности пользователей в сфере безопасности, как правило большинство разархивировали скачанное вложение, открывали .exe файл, и тем самым становились жертвой запустив работу шифровальщика.

Также были определены алгоритмы шифрования, и рассмотрена возможность дешифрования. Выявлено что, первоначально шифровальщик пускает свои корни в командную строку, из которой дает указание куда и к чему обращаться. Интерфейсы командной строки предоставляют способ взаимодействия с компьютерными системами и являются общей функцией для многих типов платформ операционных систем. Одним из примеров интерфейса командной строки в системах Windows является cmd, который может использоваться для выполнения ряда задач, включая выполнение другого программного обеспечения. Рассмотрены все этапы движения вируса шифровальщика в системе, продемонстрированы все задетые, измененные, поврежденные файлы. Так же выделены. Проанализированы аспекты социальной инженерии при фишинговой рассылке.

Список литературы

1 Описание вируса-шифровальщика URL: https://www.anti-malware.ru/threats/virus-encoder

2 Семейство Ransomware URL: https://www.malwarebytes.com/ransomware/

3 Шифрование и вирус URL: https://hackernoon.com/cryptography-malware-ransomware-36a8ae9eb0b9

4 Вспомогательный ресурс для перевода URL: https://yandex.kz/search/?lr=162&text=self%20

5 Шифрование URL: http://comp-neo.ru/virus-shifrovalshchik-rasshifrovat-fajly

6 Методы заражения URL: http://comp-neo.ru/virus-shifrovalshchik-rasshifrovat-fajly

7 Способы борьбы URL: https://www.kaspersky.ru/blog/ransomware-blocker-to-cryptor/12301/

8 Нормы микроклимата URL: http://adilet.zan.kz/rus/docs /V050003789

9 Аманжолова К. Б., Алибаева С. А. Экономика предприятии телекоммуникации: Учебное пособие. - Алматы: АИЭС, 2003

10 Голубицкая Е. А., Жигульская Г. М. Экономика связи. – М.: Радио и связь, 2000г.

Список сокращений

1. ПО – программное обеспечение.
2. ОС – операционная система.
3. ОФ – основные фонды.
4. РК – Республика Казахстан.
5. ПК – персональный компьютер.