

# Борьба с фишингом при помощи ClamAV



**Сергей Яремчук**

У пользователей и администраторов свободный антивирус ClamAV [1] ассоциируется в первую очередь с борьбой с вирусами, но относительно недавно он научился определять и фишинг-адреса.

Сегодня по результатам некоторых рейтингов [2] ClamAV является одним из наиболее популярных Open Source-проектов, что, впрочем, и неудивительно. Ведь это единственный проект, обеспечивающий нужную функциональность. О возможностях ClamAV уже говорилось в журнале [3, 4], большая часть информации действительна до сих пор, но кое-что уже изменилось. В частности, в указанных источниках отсутствует информация по теме статьи.

Проект ClamAV появился во времена рассвета «популярности» почтовых червей. И основной его задачей была разработка такой программы, которую можно было легко интегрировать с максимально большим количеством почтовых серверов, работающих под UNIX. То есть именно проверка электронных сообщений была и до сих пор остается приоритетным направлением его развития. Поэтому интерес к борьбе с фишингом далеко не случаен.

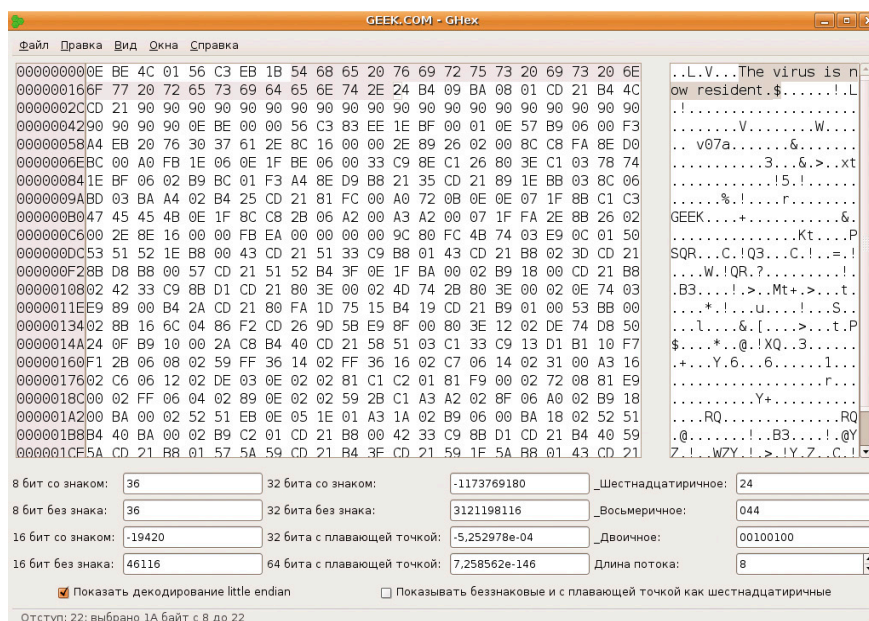
Первые записи о включении функций борьбы с фишингом в Changelog датированы январем 2005 года. Некоторое время было затрачено на тестирование и устранение недостатков. Но основные изменения появились после того, как в рамках программы Google Summer of Code 2006 был разработан новый формат сигнатур, обеспечивающий большую эффективность по сравнению со старыми. А новый модуль Advanced phishing detection, появившийся в CVS в сентябре 2006 года, хотя пока и имеет статус experimental, но уже умеет сверять указанный в сообщении URL с реальным сайтом. В это же время добавлены и специальные параметры в конфигурационный файл clamd.conf. И наконец, начиная с версии 0.90, увидевшей свет в начале 2007 года, модуль проверки на фишинг входит в ClamAV стандартно. Для включения экспериментального кода, обеспечивающего эвристический анализ сообщений, следует добавить параметр --enable-experimental при конфигурировании.

## Параметры запуска

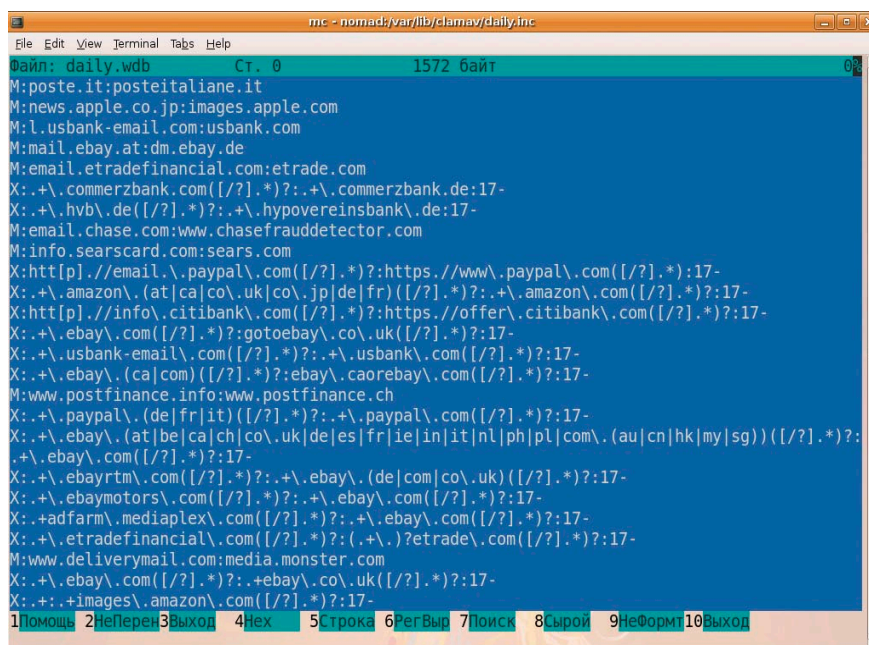
За антифишинг-проверку в конфигурационном файле демона clamd.conf отвечает несколько параметров:

```
# Разрешаем проверку сообщений на основании сигнатур
PhishingSignatures yes
```

```
# Разрешение эвристического анализа URL
PhishingScanURLs yes
```



Авторы некоторых вирусов оставляют комментарии, которые можно использовать при построении сигнатур



WDB-файл содержит список узлов, которые не будут блокироваться

```
# Работа только с доменами, указанными в .pdb базе данных.
# Сканирование всех доменов увеличивает количество ложных
# позитивных срабатываний (то есть определяет нормальных
# сообщений как фишинг)
PhishingRestrictedScan yes
```

```
# Всегда блокировать скрытые HTTPS в URL, даже если URL
# не находится в базе данных. Его использование может
# привести к увеличению ложных срабатываний
PhishingAlwaysBlockSSLMismatch no
```

```
# Всегда блокировать маскируемый URL
PhishingAlwaysBlockCloak no
```

Эти параметры действительны для версии ClamAV 0.93. Параметры PhishingSignatures и PhishingScanURLs установлены в «Yes» по умолчанию. Поэтому даже если в конфигурационном файле их нет, проверка на фишинг активизирована.



Антифишинг-технология пока находится в стадии активной разработки, поэтому вполне вероятно, что со временем эти параметры могут быть заменены другими или переименованы. Например, ранее параметр PhishingRestrictedScan назывался PhishingScanAllDomains, а чуть позже – Phishing StrictURLCheck.

Утилиты clamscan и clamdscan также имеют параметры, соответствующие указанным выше. Так, чтобы отключить антифишинг-проверку при сканировании обычных файлов, следует использовать параметры no-phishing-sigs и no-phishing-scan-urls:

```
$ clamscan --no-phishing-sigs -r
--no-phishing-scan-urls -r
-r /home/grinder/temp
```

Для активации проверки всех доменов (то есть PhishingRestricted Scan no) следует использовать --no-phishing-restrictedscan. И наконец, еще два параметра --phishing-ssl и --phishing-cloak отключают блокировку скрытых SSL и URL.

## Все дело в базах

Как видно из настроек, ClamAV поддерживает два метода определения фишинг-адресов: эвристический и на основании сигнатур.

Первый режим, как уже говорилось, обеспечивается специальным модулем. Соответственно и запись в журнале в зависимости от того, какой из методов сработал, будет иметь разный вид.

Например, в случае использования эвристического анализатора:

```
Phishing.Heuristics.Email.SpoofedDomain URL domain mismatch
Phishing.Heuristics.Email.HexURL suspicious hexadecimal URL notation
```

И сигнатурного:

```
Email.Phishing.Card Mail (type 4) Credit Cards (Access/Visa etc.)
HTML.Phishing.Card HTML (type 3) Credit Cards (Access/Visa etc.)
```

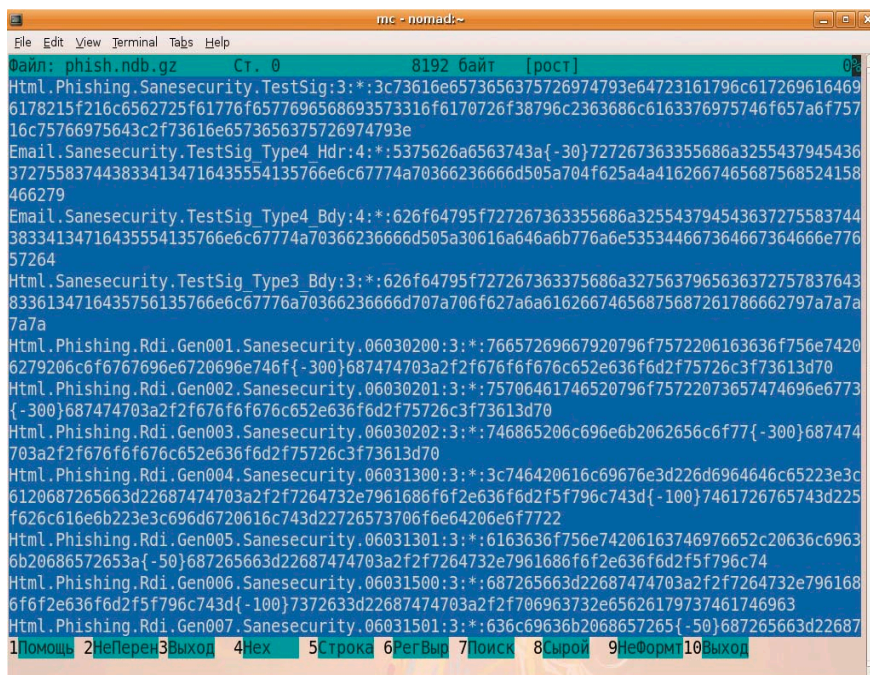
Записи об известных фишинг-адресах сохраняются в антивирусных базах и обновляются вместе с остальными сигнатурами.

В ClamAV используется две базы: main и daily. Первая – постоянная, вторая – для ежедневных обновлений. Их расположение можно узнать из переменной DatabaseDirectory, конфигурационного файла clamd.conf (в Ubuntu – /var/lib/clamav).

При установке из репозитория базы, как правило, находятся в каталоге в распакованном виде.

```
$ ls /var/lib/clamav/daily.inc/
```

```
COPYING daily.fp daily.hdu daily.mdb daily.ndu daily.zmd
daily.cfg daily.ftm daily.ign daily.mdu daily.pdb
daily.db daily.hdb daily.info daily.ndb daily.wdb
```



Неофициальные сигнатуры поставляются в расширенном NDB-формате

При установке из исходных текстов по умолчанию базы находятся в подкаталоге /usr/share/clamav, сюда будут помещены два файла main.cvd и daily.cvd, которые являются упакованными CVD (ClamAV Virus Database)-базами. Как их распаковать в случае необходимости, рассказано во врезке «Добавление сигнатуры вируса».

Антифишинга сигнатур, идущих в ClamAV, может оказаться недостаточно. Например, при сканировании большинства тестовых сообщений, взятых с сайтов [6, 7], присутствие фишинг-составляющих не обнаруживается. Кстати, на www.millersmiles.co.uk в качестве примера используются реальные письма. Поэтому стоит обратить внимание на дополнительные базы.

Например, на сайте [5] предлагаются три базы:

- **phishbar** – содержит информацию о графических объектах, применяемых на фишерских сайтах;
- **phish** – фишинг-сигнатуры;
- **scam** – сигнатуры сообщений, используемых мошенниками, предлагаемые различного рода лотереи, акции и прочие аферы 419.

Кроме этого на сайте доступно несколько вариантов скриптов для автоматического обновления указанных баз.

## Форматы баз

Данные, имеющие отношение к борьбе с фишингом, находятся в двух базах: WDB и PDB, каждая из которых имеет свой собственный формат.

Подробнее процесс создания своих записей описан в документе «Phishing signatures creation HOWTO», доступном в архиве с исходными текстами.

В WDB (White DataBase)-базу заносятся разрешенные пары URL, при совпадении с которыми дальнейшей проверки не будет, а в PDB (Phishing DataBase) содержатся имена или адреса известных фишинг-узлов.

Файл PDB может содержать запись в одном из двух форматов H (Host) или R (Regular).

Запись вида H несколько проще, так как здесь описывается домен или субдомен, без применения регулярно выражения:

```
H[Filter]:DisplayedHostname |
[:FuncLevelSpec]
```

Например:

```
H:bankofamerica.com
```

Под эту запись попадут и все поддомены bankofamerica.com, но сайт с адресом bankofamerica.com.uk пройдет через фильтр.

Дополнительные Filter описаны в файле libclamav/phishcheck.h. В версии 0.93 запись выглядит так:

```
#define CHECK_SSL 1
#define CHECK_CLOAKING 2
#define CLEANUP_URL 4
#define CHECK_IMG_URL 8

#define CL_PHISH_ALL_CHECKS |
(CLEANUP_URL|CHECK_SSL|
CHECK_CLOAKING|CHECK_IMG_URL)
```

В документации приведено больше вариантов, но, очевидно, ориентироваться в первую очередь следует на содержимое phishcheck.h. По умолчанию все проверки, указанные в Filter, включены, что является наиболее разумным решением, изменять такое поведение следует лишь в очень редких случаях.

Параметр FuncLevelSpec указывает на функциональный уровень. Этот параметр, появившийся в версии 0.70, позволяет установить соответствие версии движка антивируса и сигнатур базы данных. Об их несовпадении говорит такая запись:

```
WARNING: Current functionality level = 14, recommended = 26
```

Сигнатуры, не удовлетворяющие текущему уровню движка, не будут считываться и использоваться. Текущую версию functionality level можно узнать несколькими способами. Самый простой анализ журналов freshclam.log:

```
$ ls /var/lib/clamav/daily.inc/
```

```
main.inc is up to date (version: 46, sigs: 231834, f-level: 26, builder: sven)
daily.inc updated (version: 7231, sigs: 69743, f-level: 26, builder: ccoodes)
```

В данном примере движок антивируса соответствует 26 functionality level (f-level: 26).

В большинстве случаев в FuncLevelSpec можно ничего не писать. Хотя учитывая, что нормальная поддержка регулярных выражений появилась начиная с f-level – 17, в записях вида R можно использовать значение «17».

## Добавление сигнатуры вируса

Так как информация по самостоятельному добавлению в базу сигнатуры вируса, который не обнаруживается в ClamAV, описанная в [3], несколько устарела, то скажу пару слов о том, как это сделать. В архиве исходных текстов есть документ «Creating signatures for ClamAV», в котором описано, как получить сигнатуру из тестовых вирусов, поставляемых вместе с ClamAV. Для работы с базами используется утилита sigtool. Чтобы распаковать CVD-базу, ее следует запустить с ключом --unpack:

```
$ sigtool --unpack=daily.cvd
```

Самый простой способ создать сигнатуру – записать его MD5-сумму:

```
$ sigtool --md5 test.exe > test.hdb
```

Смотрим, что внутри:

```
$ cat test.hdb
```

```
9e48e83d9b05738b571720401a169263:533:test.exe
```

Теперь проверяем, подключив новую базу:

```
$ clamscan -d test.hdb test.exe
```

```
test.exe: test.exe FOUND
```

Вирус определился, но стоит ему только заразить другой файл, как данная схема будет неэффективна. В этом случае необходимо сохранить в базу специфическую часть, хотя, конечно, это потребует некоторого опыта. Чтобы получить дамп, нужно добавить параметр --hex-dump.

```
$ cat test.exe | sigtool |
--hex-dump > virus.sig
```

Или воспользоваться Hex-редактором

вроде hexedit или GHex. Подойдет и файловый менеджер Midnight Commander. Авторы некоторых вирусов оставляют комментарии, которые можно найти при помощи утилиты string, а потом перевести в hex. Запись в базе вирусов ClamAV в самом простом случае выглядит так: «имя вируса:сигнатура».

Можно создать отдельную базу, которую подключить при помощи ключа --database или скопировав в DatabaseDirectory. Как вариант занести новые сигнатуры в daily базу данных.

Добавляем в одну из баз сигнатуру и просчитываем новую контрольную сумму измененного файла:

```
$ cat virus.sig >> daily.db
$ md5sum daily.db >> daily.info
```

После чего удаляем предыдущую запись о MD5-сумме файла daily.db и правим daily.info, чтобы запись выглядела так:

```
daily.db: 56c2da1eb6778c6eaa19660e119d46cd
```

Теперь можно пользоваться обновленной базой.

Кстати, начиная с версии 0.91.2, в ClamAV введена возможность определения Potentially Unwanted Applications (потенциально нежелательных приложений, PUA). База PUA содержит сигнатуры приложений, которые сами по себе не вредоносны, но могут быть использованы с такой целью. Для активации такого режима следует использовать параметр командной строки --detect-pua или «DetectPUA yes». Хотя пока его использование на промышленных системах не рекомендуется самими разработчиками.

Запись вида R имеет такую форму:

```
R[Filter]:RealURL:DisplayedURL[:FuncLevelSpec]
```

В поле RealURL содержится регулярное выражение, соответствующее реальному имени узла, а в DisplayedURL – которое выводится пользователю. Проще говоря:

```
<a href="http://RealURL"> DisplayedURL </a>
```

Например:

```
X:.\.amazon\.(at|ca|co.uk|co.jp|de|fr)([/\?].*)?:.\.amazon\.com([/\?].*)?:17-
```

В данном примере описаны сайты с доменом RealURL = amazon.at, amazon.ca и так далее, которые пользователю показываются как amazon.com.

В WDB-файле формат записей схож.

```
X:RealURL:DisplayedURL[:FuncLevelSpec]
```

## Фишинг

Термин «фишинг» (phishing) созвучен английскому fishing – рыбалка, удить. Он произошел от слияния трех слов: password (пароль), harvesting (сбор) и fishing: то есть означает ловлю и сбор паролей. Мошенники от имени популярных или вызывающих доверие веб-узлов рассылают миллионы фишинг-сообщений в виде спама. Электронные сообщения, всплывающие окна и веб-узлы, на которые даны ссылки, выглядят официально, а дизайн сайтов полностью идентичен оригинальным, что позволяет мошенникам обманывать многих людей, заставляя поверить, что письмо действительно получено от надежной организации. Но ни о чем не подозревающие пользователи часто отвечают на требования мошенников предоставить номера кредитных карточек, пароли, сведения о банковском счете или другие личные данные. Делается это под предлогом разблокирования, подтверждения, обновления данных учетной записи, восстановления пароля, тестирования нового сервиса или новых возможностей (например, прямой перевод де-

нег в другую систему), других плановых мероприятий. При этом используются разные методы маскировки URL, чтобы сделать их более похожими на оригинальные.

Начав с бесплатных хостингов и адресов типа bank.mail.ru:

```
<a href=" bank.mail.ru ">bank.ru</a>
```

Мошенники совершенствовали свои технологии, используя адреса сайтов, записанных перед знаком @, то есть http://www.bank.ru@12345.com. По спецификации то, что записано перед знаком @, считается данными пользователя сайта, записанного после @. Это значит, что фактически обращение идет к сайту 12345.com. В последних версиях Internet Explorer такая адресация запрещена, а в Firefox она вызывает специальное предупреждение, но несколько лет назад эта уловка отличалась работала.

Есть и другие варианты: использование IP-адреса вместо имени или похожего имени сайта (banc.ru), маскировка адреса путем его кодирования в шестнадцатичисловое число.


- нормализация URL (убираются лишние символы, символы «\» заменяются на «/», адрес приводится к верхнему регистру и так далее);
- проверяется адрес в WDB, при обнаружении проверка заканчивается;
- извлекается домен и имя узла, которые также сверяются с WDB;
- проверяется наличие закодированного IP-адреса;
- проверяется наличие SSL в RealURL (если DisplayedURL использует http, а в RealURL – https, то заблокировать такие ссылки можно, установив PhishingAlwaysBlockSSL Mismatch в yes);
- если на месте RealURL вместо имени стоит IP-адрес, такой узел блокируется;
- далее проверяется соответствие в RealURL и DisplayedURL сначала имени узла, а затем домена, при совпадении проверка заканчивается.

В настоящее время отключена возможность использования службы

DNS для проверки совпадения имени и IP-адреса, так как это может привести к большому количеству запросов.

## Заключение

Фишинг – это реалии сегодняшнего Интернета, и с этим явлением приходится считаться и бороться. Появление в ClamAV возможности определять и блокировать такие сообщения, позволяющие повысить защиту пользователей, только приветствуется. Конечно, пока данная разработка находится еще в стадии активного развития, но все же она уже позволяет блокировать часть писем, рассылаемых фишерами.

Удачи! 

```
M:RealHostname:DisplayedHostname[:FuncLevelSpec]
```

Как и в случае с PDB, в X-формате используются регулярные выражения, а в M – имена узлов. Например:

```
X:.\.paypal\. (de|fr|it) ([/?].*)?:.\.paypal\.com ([/?].*)?:17-M:news.apple.co.jp:images.apple.com
```

В первом примере имеем зеркала сайта RealHostname – paypal.de, paypal.fr и paypal.it, которые пользователи выводятся с более известным именем paypal.com (DisplayedHostname).

Базы с www.sanesecurity.com поставляются в расширенном формате NDB (Extended DataBase), который применяется для описания сигнатур вирусов. В общем, формат его таков:

```
MalwareName:TargetType:Offset:HexSignature _[:MinEngineFunctionalityLevel:[Max]]
```

Назначения полей понятны из названий. Поле TargetType указывает на тип целевого файла, в котором проверяется наличие этой сигнатуры. В сигнатурах www.sanesecurity.com здесь обычно используются цифры 3 (HTML) или 4 (mail).

## Порядок проверки

При проверке любого адреса на принадлежность к фишинг-ресурсу будут пройдены шаги, которые описаны в libclamav/phishcheck.c. Совпадение с любым из них, как правило, приводит к завершению проверки текущего адреса:

- проверяется совпадение RealURL=DisplayedURL, если совпадает, то проверка заканчивается;

1. Сайт проекта ClamAV – <http://www.clamav.net>.
2. Top 75 Open Source Security Apps – [http://www.esecurityplanet.com/article.php/11162\\_3741146\\_1](http://www.esecurityplanet.com/article.php/11162_3741146_1).
3. Яремчук С. Свободный антивирус. //Системный администратор, № 3, 2004 г. – С. 32-37 – [http://www.samag.ru/art/03.2004/03.2004\\_06.pdf](http://www.samag.ru/art/03.2004/03.2004_06.pdf).
4. Супрунов С. Еще раз о ClamAV: особенности установки во FreeBSD. //Системный администратор, № 8, 2004 г. – С. 24-25.
5. Неофициальные антифишинг-сигнатуры – <http://www.sanesecurity.com/clamav/downloads.htm>.
6. Сайт, посвященный борьбе с фишингом, – <http://www.millersmiles.co.uk>.
7. Сайт, посвященный афере 419, – <http://www.nigerian419scams.com>.
8. Еще один полезный ресурс с постоянно обновляющейся информацией о фишинг-сайтах – <http://www.phishtank.com>.