ataques super eficientes de Denial of Service

Jan Seidl

$ whoami

Full Name: Jan Seidl

Origin: Rio de Janeiro, RJ - Brazil

Work:

- **Technical Coordinator @ TI Safe**
- **OpenSource contributor for: PEV, Logstash**
- **Codes and snippets @ github.com/jseidl**

Features:

- **UNIX Evangelist/Addict/Freak (but no fanboy!)**
- **Python and C lover**
- **Coffee dependent**
- **Hates printers and social networks**
- **OctaneLabs Forensic Group Member**

agenda

0x0 Introdução à Negação de Serviço
0x1 Background: Ataques Layer 3
0x2 Atacando a Layer 7: Fundamentos
0x3 Atacando a Layer 7: Vetores e Tools
0x4 WebServer DoS Mitigation 101
0x5 Proxies (SOCKS/TOR) e ataques Layer 7
0x6 Jericho Attack Technique: Load-balancing attacks
0x7 XSS D/DoS
0x8 Size doesn't matter: Mobile-launched Denial-of-Service
0x9 Demo/Video: GoldenEye MdoS Android Tool
0xA Dúvidas?

Introdução à Negação de Serviço

O que é Negação de Serviço?

Introdução à Negação de Serviço

O que é Negação de Serviço?

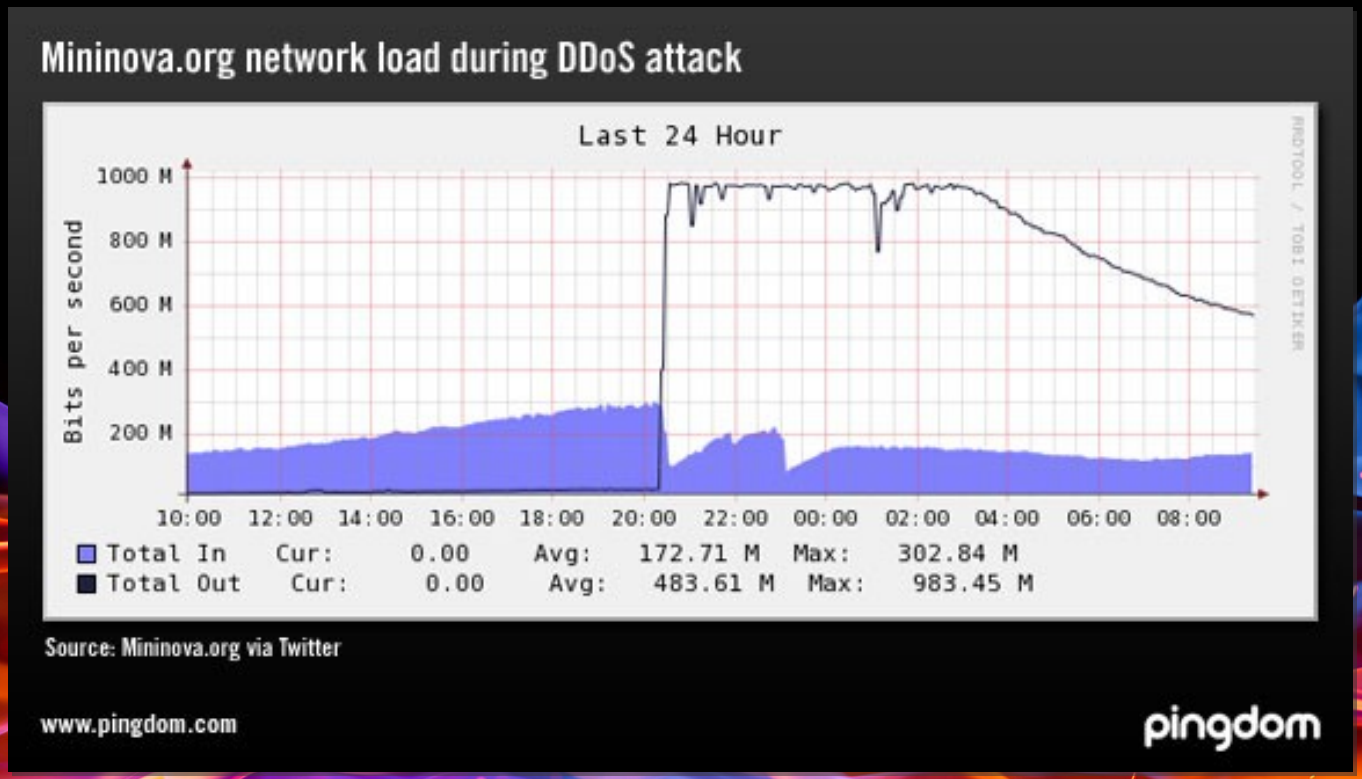
Um ataque de negação de serviço (...), é **uma tentativa em tornar os recursos de um sistema indisponíveis para seus utilizadores.**

Fonte: Wikipedia/pt_BR

Introdução à Negação de Serviço

Introdução à Negação de Serviço

Resultado?



Introdução à Negação de Serviço

Resultado?

Introdução à Negação de Serviço

Introdução à Negação de Serviço

Sintomas

Performance estranhamente lenta

Indisponibilidade de determinado recurso

Indisponibilidade de todos os recursos

Introdução à Negação de Serviço

Casos Recentes

Introdução à Negação de Serviço

CNNMoney FORTUNE

Home | Video | Markets | Investing | Ec

Apple 2.0 | Big Tech | Tech Tumblr | Innovation Nation | Startups | Brainstorm Tech

THE CYBERCRIME ECONOMY

Major banks hit with biggest cyberattacks in history

CNNMoney

comments

By David Goldman @CNNMoneyTech September 28, 2012: 9:27 AM ET

NEW YORK (CNNMoney) -- There's a good chance your bank's website was attacked over the past week.

Since Sept. 19, the websites of Bank of America (**BAC**, **Fortune 500**), JPMorgan Chase (**JPM**, **Fortune 500**), Wells Fargo (**WFC**, **Fortune 500**), U.S. Bank (**USB**, **Fortune 500**) and PNC Bank have all suffered day-long slowdowns and been sporadically unreachable for many customers. The attackers, who took aim at Bank of America first, went after their targets in sequence. Thursday's victim, PNC's website, was inaccessible at the time this article was published.

http://money.cnn.com/2012/09/27/technology/bank-cyberattacks/index.html

Introdução à Negação de Serviço

SECURITY dark READING
Protect The Business Enable Access

REPORT: **Bypassing the IPS**
Find out how attackers can avoid intrusion prevention systems and gain access to you data. Download now!

Welcome Guest. | Log In | Register | Membership Benefits

ATTACKS / BREACHES | VULNERABILITIES | APPLICATION SECU
SECURITY MANAGEMENT | STORAGE SECURITY | ENCRYPTION | NAC

Tech Center: Advanced Threats

Tweet 105 Curtir 36 Share +1 Permalink
BOOKMARK ...

New Denial-Of-Service Attack Cripples Web Servers By Reading Slowly

'Slow Read' proof-of-concept and tool released Thursday

Jan 05, 2012 | 03:51 PM | 0 Comments

By **Kelly Jackson Higgins**
Dark Reading

A researcher today published proof-of-concept code that takes a different spin on the slow HTTP denial-of-service (DoS) attack simply by dragging out the process of reading the server's response -- and ultimately overwhelming it.

Introdução à Negação de Serviço

http://www.datacenterknowledge.com/archives/2009/08/06/twitter-is-latest-victim-in-series-of-attacks/

Introdução à Negação de Serviço

nakedsecurity

Award-winning news, opinion, advice and research from SOPHOS

malware mac facebook android vulnerability data loss privacy more...

◀ SSCC 87 - Mac botnet, Global Paym… Twitter hauls spammers into court ▶

Anonymous attacks UK Prime Minister and Home Office websites with DDoS assault

by Graham Cluley on April 7, 2012 | 20 comments

FILED UNDER: Denial of Service, Featured, Law & order

Anonymous hacktivists have launched a distributed denial-of-service attack against the websites of 10 Downing Street and the British government's Home Office website, preventing legitimate users from visiting the sites by flooding them with unwanted internet traffic.

http://nakedsecurity.sophos.com/2012/04/07/anonymous-attacks-home-office/

Introdução à Negação de Serviço

USA TODAY | Tech Subscribe | Mobile Google USA TODAY stories, photos and more

Home | News | Travel | Money | Sports

Tech: Blogs | Products | Gaming | Science & Space | TV on the Web

Hacktivist attacks grow as governments get in on the action

By Byron Acohido, USA TODAY

LAS VEGAS – Denial-of-service attacks are surging all across the Internet.

By Alejandro Gonzalez, USA TODAY

Denial-of-service attacks surged 70% in the first six months of 2012, compared to the same period of 2011.

Individuals and groups have perfected the art of knocking websites offline for hours, days or sometimes much longer.

Such attacks surged nearly 70% in the first six months of 2012 vs. the same period in 2011, according to statistics released exclusively to USA TODAY by Prolexic, a Hollywood, Fla.-based website defense firm.

And it's not just the usual suspects who are responsible. The attacks increasingly have a geopolitical bent.

Two prominent global hacktivist collectives — Anonymous and LulzSec — gained notoriety over the past two years for disrupting the Web presence of scores of corporations and government agencies, then bragging about it on Twitter, YouTube and Facebook. Their standing motive: Mete out punishment for perceived bad corporate practices and unfair government policies.

http://usatoday30.usatoday.com/tech/news/story/2012-07-19/hactivism-anonymous-attacks/56464792/1

Introdução à Negação de Serviço

http://olhardigital.uol.com.br/negocios/digital_news/noticias/ataques-ddos-cresceram-70-em-2012,-diz-pesquisa

Introdução à Negação de Serviço

Alvos (Na camada OSI)

Network (Layer 3)
Consumo de banda / Redução de performance da banda

Application (Layer 7)
Consumo de recursos da aplicação ou OS

Introdução à Negação de Serviço

Network (Layer 3)

Consumo de banda / Redução de performance da banda

Background: Ataques Layer 3

Ataques populares

Ping Flood

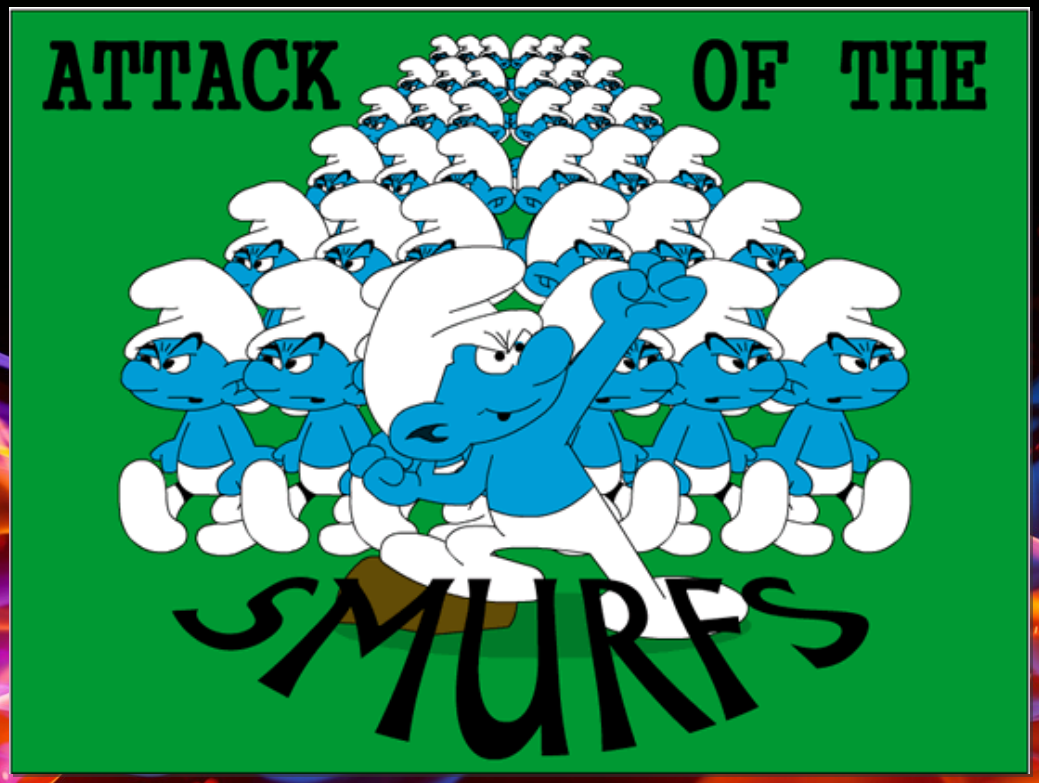
(…) é um ataque (...) no qual o atacante sobrecarrega o sistema vítima com pacotes ICMP Echo Request (pacotes ping). (...) Como a vitima tentará responder aos pedidos, irá consumir a sua largura de banda impossibilitando-a responder a pedidos de outros utilizadores.

Fonte: Wikipedia/pt_BR

Background: Ataques Layer 3

Ataques populares

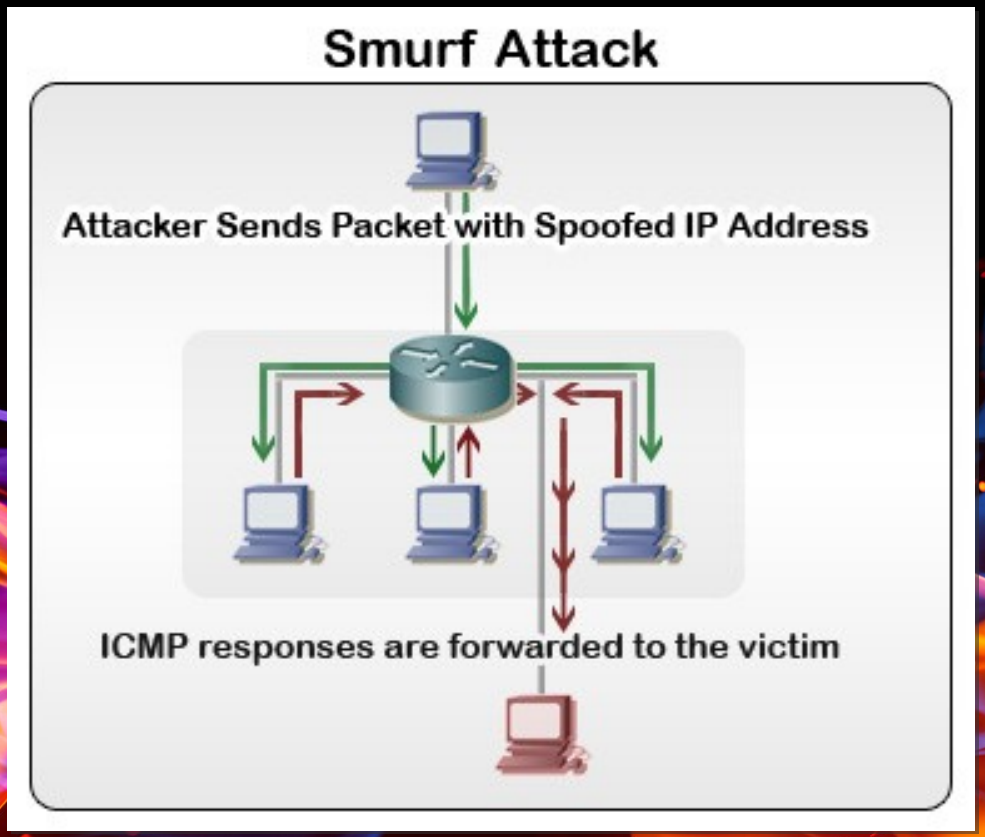
Smurf Attack



Background: Ataques Layer 3

Ataques populares

Smurf Attack



Background: Ataques Layer 3

Ataques populares

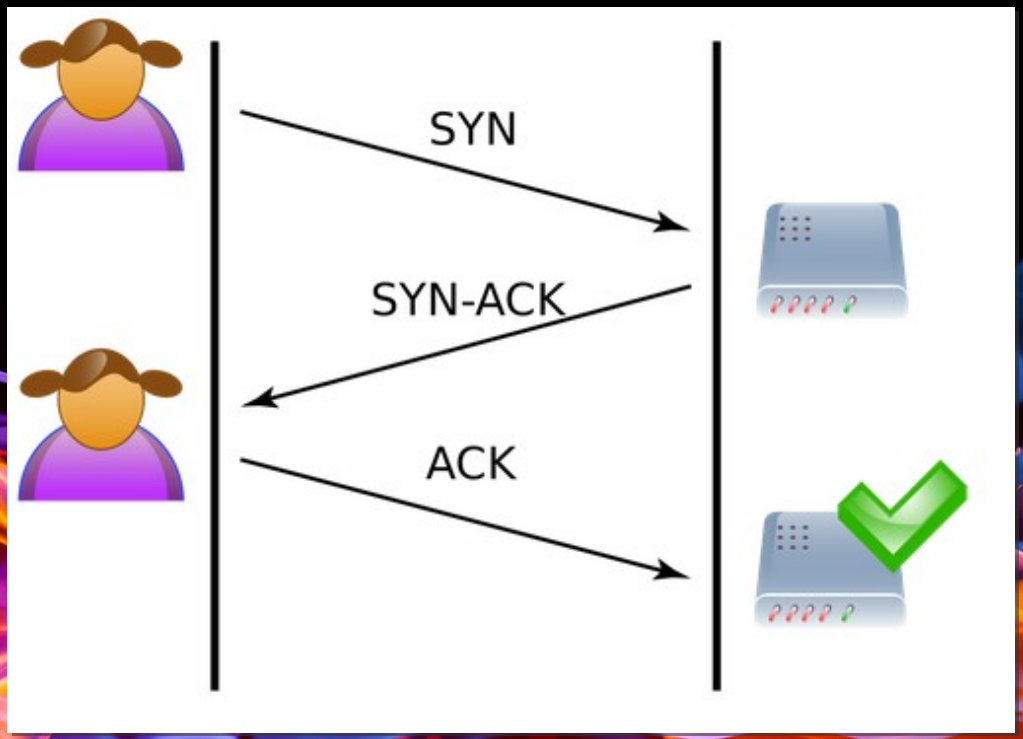
Smurf Attack

<deprecated>

Background: Ataques Layer 3

Ataques populares

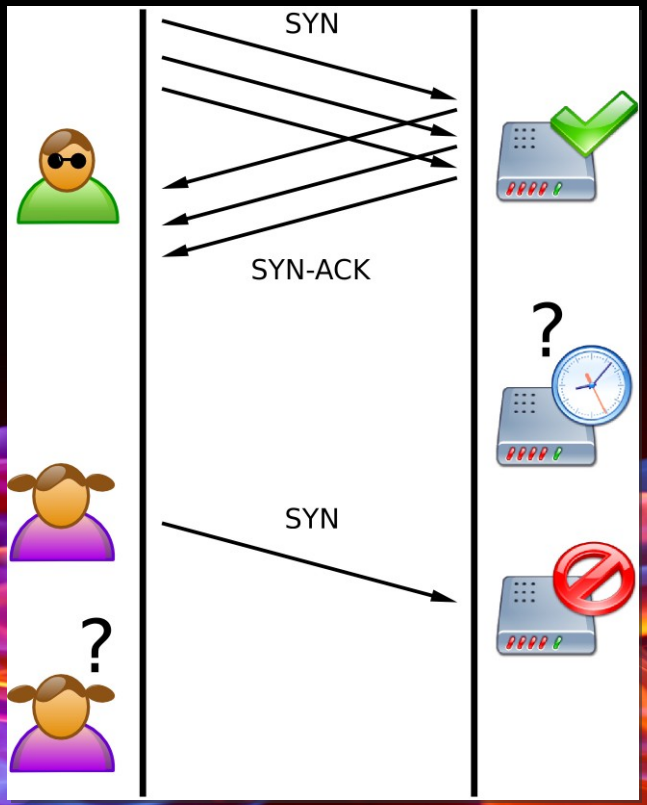
SYN Flood



Background: Ataques Layer 3

Ataques populares

SYN Flood



Background: Ataques Layer 3

Ataques populares

Teardrop Attack

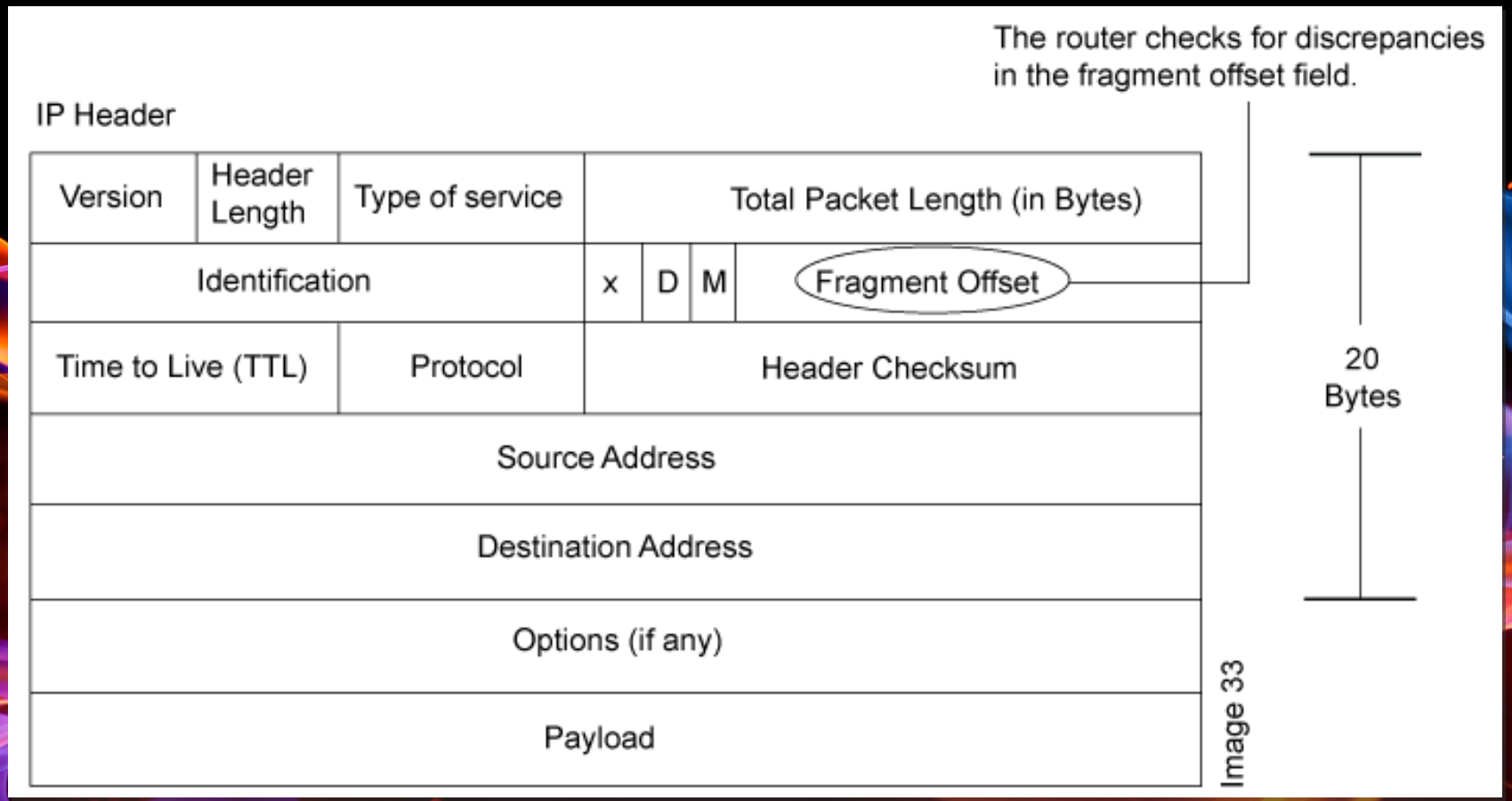
“When the sum of the offset and size of one fragmented packet differ from that of the next fragmented packet, the packets overlap, and the server attempting to reassemble the packet can crash, especially if it is running an older operating system that has this vulnerability.”

http://www.juniper.net/techpubs/software/junos-es/junos-es93/junos-es-swconfig-security/understanding-teardrop-attacks.html

Background: Ataques Layer 3

Ataques populares

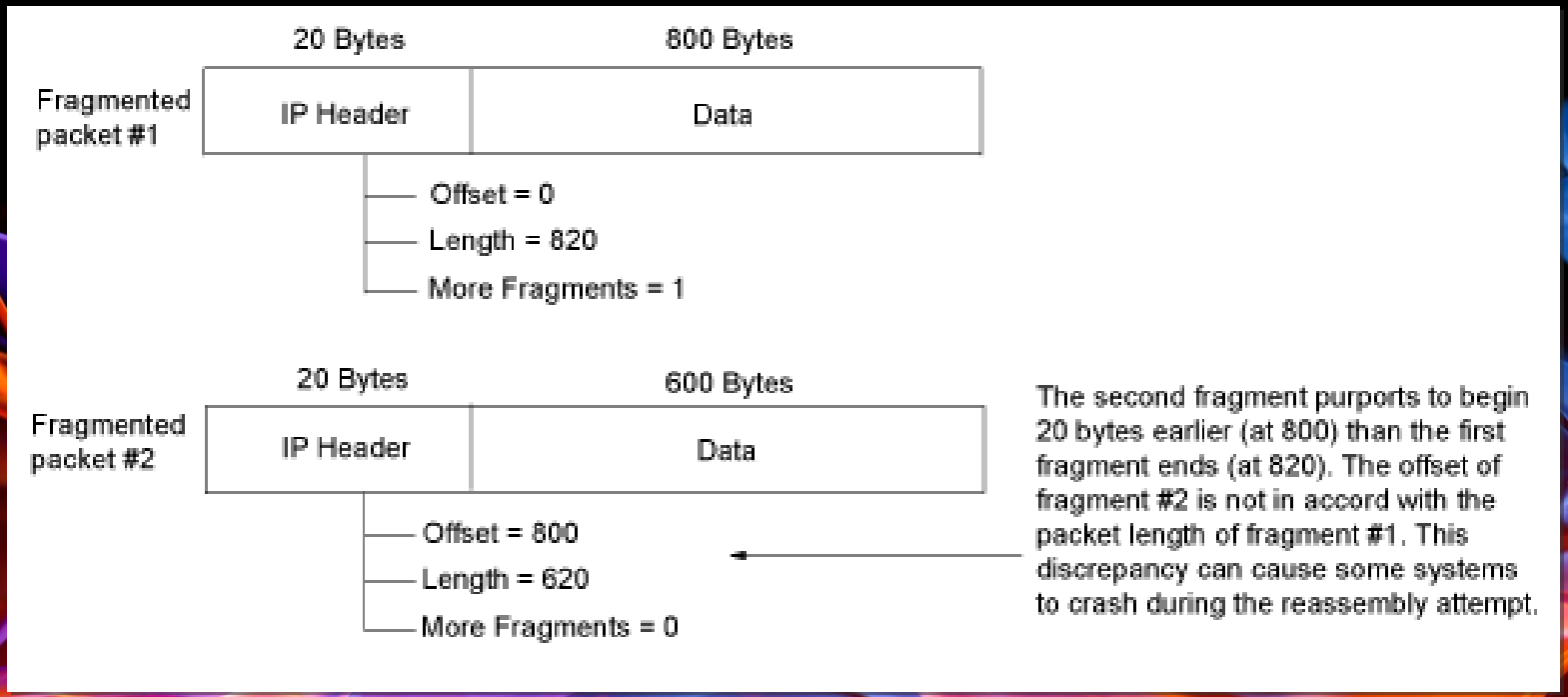
Teardrop Attack



Background: Ataques Layer 3

Ataques populares

Teardrop Attack



Background: Ataques Layer 3

Ataques populares

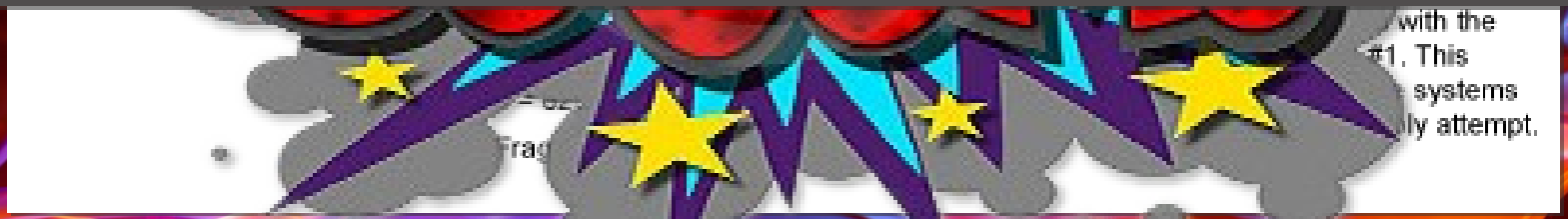
Teardrop Attack

Background: Ataques Layer 3

Ataques populares

Teardrop Attack

<deprecated>



Atacando a Layer 7: Fundamentos

Application (Layer 7)

Consumo de recursos da aplicação ou OS

Atacando a Layer 7: Fundamentos

Foco

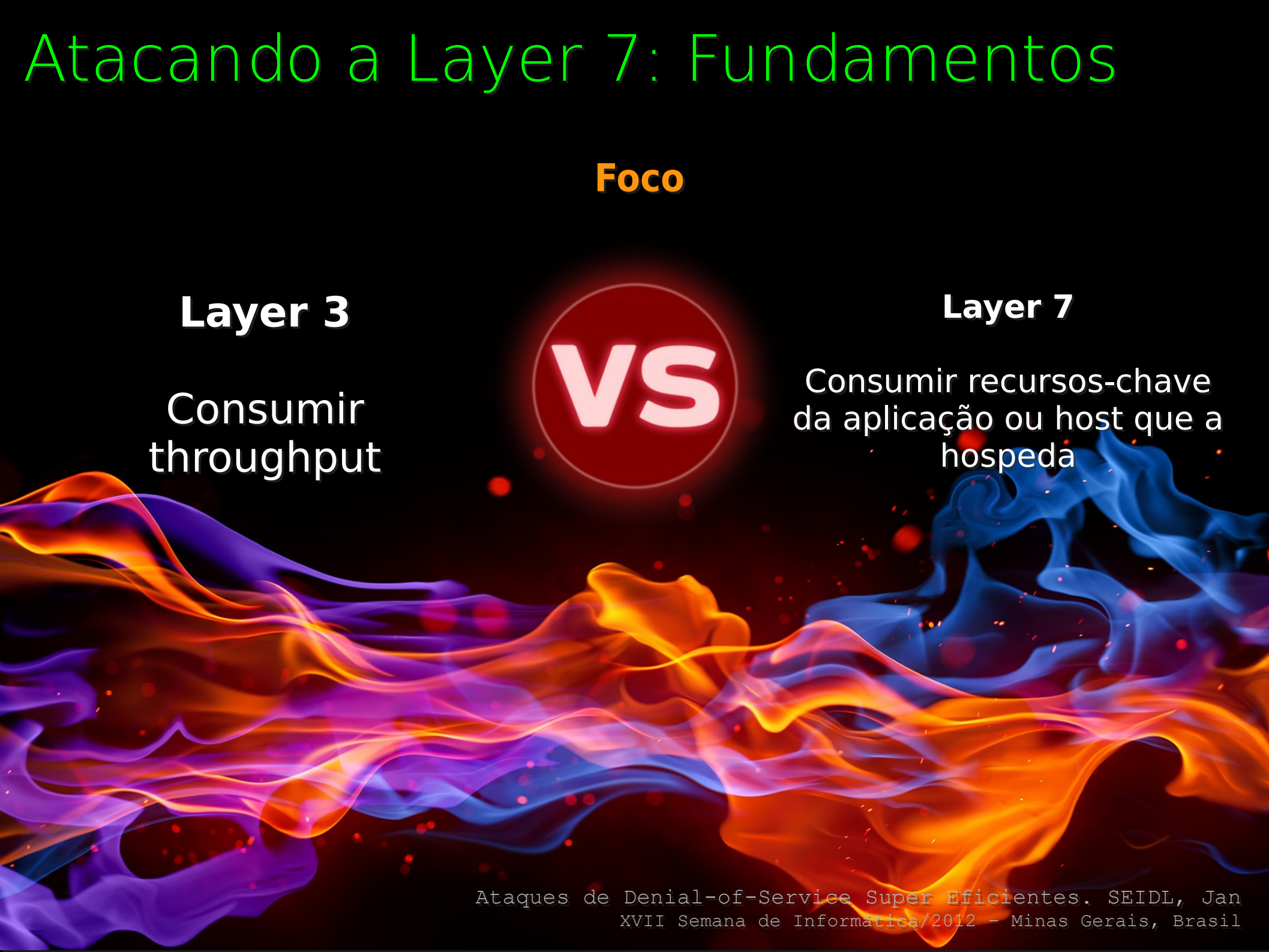
Layer 3

Consumir throughput

VS

Layer 7

Consumir recursos-chave da aplicação ou host que a hospeda



Atacando a Layer 7: Fundamentos

Stealthness

Layer 3

Alto ruído na rede
(noisy attack)

Layer 7

Baixo ruído na rede, pode emular requests legítimos

Atacando a Layer 7: Fundamentos

Eficiência

Layer 3

Requer muitos participantes para gerar outages consideráveis

Layer 7

Requer apenas uma máquina para ser bem sucedido

Atacando a Layer 7: Fundamentos

Contenção

Layer 3

Link largo, connection-limiting, rate-limiting

Layer 7

Atacando a Layer 7: Fundamentos

Alvos do ataque em layer 7

Operações de uso intenso de CPU, Disk I/O & Swapping, Queries longas e/ou complexas.

Recursos finitos da aplicação: Limites de Sockets Máximos, Memória Máxima, Espaço em Disco etc

Atacando a Layer 7: Vetores e Tools

Atacando a Layer 7: Vetores e Tools

Uso intenso de CPU

SSL Renegotiation / SSL Handshake Attack

Necessário 15% mais de processamento no server do que no client para criptografia do handshake.

Existe desde 2003.
Ainda afeta a maioria das implementações.

Descoberto pelo grupo THC (ww.thc.org) em 2011

Atacando a Layer 7: Vetores e Tools

Uso intenso de CPU

SSL Renegotiation / SSL Handshake Attack

Ferramenta:
THC-SSL-DOS <http://www.thc.org/thc-ssl-dos/>

- ou -

```
thc-ssl-dosit() { while :; do (while :; do echo R;
done) | openssl s_client -connect 127.0.0.1:443
2>/dev/null; done }
for x in `seq 1 100`; do thc-ssl-dosit & done
```

Atacando a Layer 7: Vetores e Tools

Uso intenso de CPU

SSL Renegotiation / SSL Handshake Attack

Afeta qualquer protocolo com TLS/SSL:

HTTPS, SMTPS, POP3S, Database secure ports etc

Mitigação?

Desligar a renegociação ajuda, mas não resolve
Um acelerador SSL pode ajudar, mas também não resolve

Mitigação por IPTables
http://vincent.bernat.im/en/blog/2011-ssl-dos-mitigation.html

Atacando a Layer 7: Vetores e Tools

Uso intenso de CPU

Apache Range Header Attack

Requisita paralelamente pequenos pedaços do conteúdo de forma comprimida (gzip)

Força o webserver a realizar diversas operações de compressão paralelas e simultâneas = alto processamento.

Descoberto em 2011 (CVE-2011-3192)

Atacando a Layer 7: Vetores e Tools

Uso intenso de CPU

Apache Range Header Attack

Ferramentas:

killapache.pl <
http://seclists.org/fulldisclosure/2011/Aug/175>

Slowhttptest <http://code.google.com/p/slowhttptest/>

Atacando a Layer 7: Vetores e Tools

Uso intenso de CPU

Apache Range Header Attack

Mitigação:

SetEnvIf ou mod_rewrite

(ref: http://httpd.apache.org/security/CVE-2011-3192.txt)

Emprego de um WAF (Web Application Firewall)

Atualizar para versão 2.2.21 ou superior

Atacando a Layer 7: Vetores e Tools

Abuso dos slots de conexão

HTTP Slow Attacks

Slow Headers, Slow Post, Slow Read

Ler ou enviar dados em pequenos pedaços, com intervalos entre as leituras / escritas.

Aguardar o request completo faz parte da natureza dos Web Servers

Atacando a Layer 7: Vetores e Tools

Abuso dos slots de conexão

HTTP Slow Attacks

Slow Headers: headers da requisição de forma 'Slow'

Slow Post: campos do corpo da requisição (post data) de forma 'Slow'

Slow Read: TCP window size baixo para ler a resposta de forma 'Slow'

Atacando a Layer 7: Vetores e Tools

Abuso dos slots de conexão

HTTP Slow Attacks

Slow Headers: headers da requisição de forma 'Slow'

```
GET / HTTP/1.1 \r\n /* sleep(1) */
Connection: keep-alive \r\n /* sleep(1) */
...
```

Atacando a Layer 7: Vetores e Tools

Abuso dos slots de conexão

HTTP Slow Attacks

Slow Post: campos do corpo da requisição (post data) de forma 'Slow'

```
Content-Type: application/x-www-form-urlencoded
Content-Length: 512
Accept: text/html;q=0.9,text/plain;q=0.8
foo=bar /* sleep(1) */
bar=baz /* sleep(1) */
baz=foo /* sleep(1) */
...
```

Atacando a Layer 7: Vetores e Tools

Abuso dos slots de conexão

HTTP Slow Attacks

Slow Read: TCP window size baixo para ler a resposta de forma 'Slow'

```
/* pseudocode */
int len = 1;
while (data = read(sock, buffer, len)) {
sleep(5);
…
}
```

Atacando a Layer 7: Vetores e Tools

Abuso dos slots de conexão

HTTP Slow Attacks

Ferramentas:

Slow Headers: Slowloris, slowhttptest, OWASP HTTP Post Tool

Slow Post: RUDY, slowhttptest, OWASP HTTP Post Tool

Slow Read: slowhttptest

Atacando a Layer 7: Vetores e Tools

Abuso dos slots de conexão

HTTP Slow Attacks - Mitigação:

Slow Headers: request timeout (apache's mod_reqtimeout), WAF

Slow Post: request timeout, WAF

Slow Read: Desabilitar pipelining e window sizes anormalmente pequenos, limitar tempo máximo da conexão, WAF

Bom artigo sobre mitigação de slow attacks
https://community.qualys.com/blogs/securitylabs/2011/11/02/how-to-protect-against-slow-http-attacks

Atacando a Layer 7: Vetores e Tools

Abuso dos slots de conexão

HTTP KeepAlive + NoCache

Persiste as conexões abertas e força o webserver a regerar o conteúdo.

Primeiro POC:

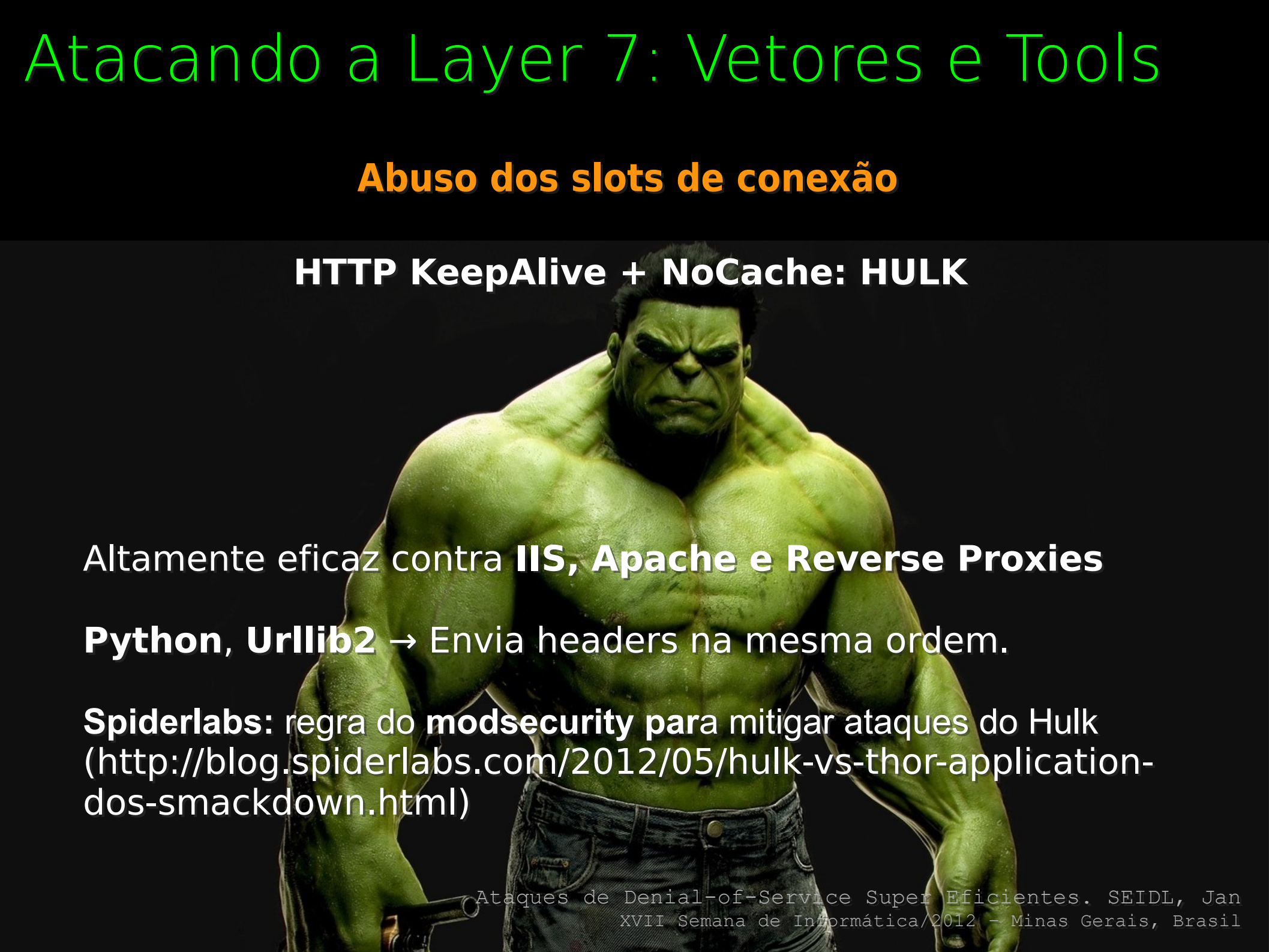
HULK – HTTP Unbearable Load King

Criado em Maio de 2012 por Barry Shteiman.
<http://www.sectorix.com/2012/05/17/hulk-web-server-dos-tool/>

Atacando a Layer 7: Vetores e Tools

Abuso dos slots de conexão

HTTP KeepAlive + NoCache: HULK



Altamente eficaz contra **IIS, Apache e Reverse Proxies**

Python, **Urllib2** → Envia headers na mesma ordem.

Spiderlabs: regra do **modsecurity para** mitigar ataques do Hulk (http://blog.spiderlabs.com/2012/05/hulk-vs-thor-application-dos-smackdown.html)

Atacando a Layer 7: Vetores e Tools

Randomização FTW!

Atacando a Layer 7: Vetores e Tools

Abuso dos slots de conexão

HTTP KeepAlive + NoCache + Randomness: GoldenEye

- Autor: Eu! :)
- Inicialmente Fork do Hulk devido sua facilidade de fingerprint
- Transformado futuramente em uma outra ferramenta independente de HTTP DoS.

Feita para testar a capacidade de bloqueio de WAFs em payloads randômicas e semi-naturais

Disponível em https://github.com/jseidl/GoldenEye

Atacando a Layer 7: Vetores e Tools

Abuso dos slots de conexão

HTTP KeepAlive + NoCache + Randomness: GoldenEye

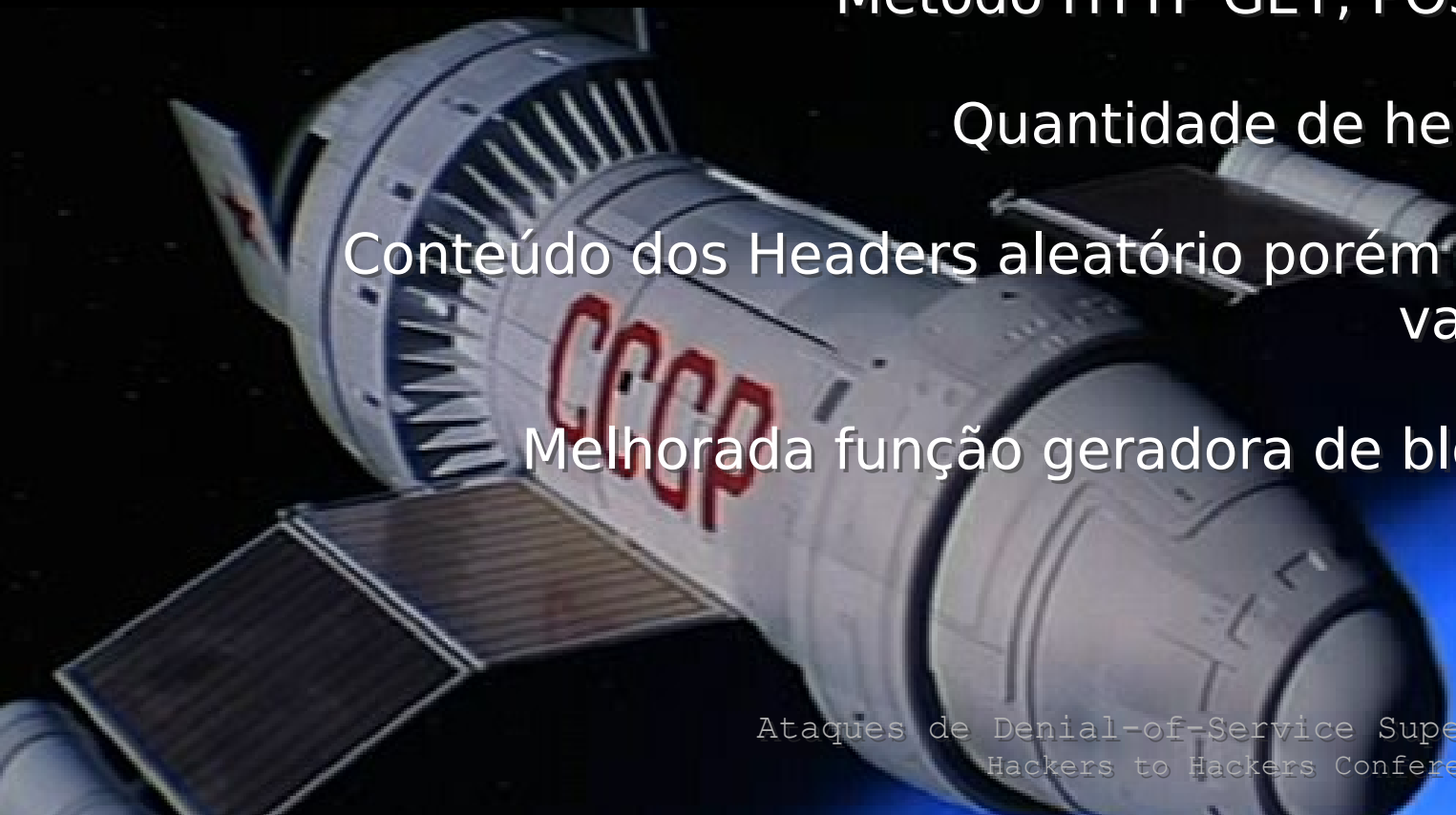
Main Features:

Método HTTP GET, POST ou Aleatório

Quantidade de headers aleatória

Conteúdo dos Headers aleatório porém coerentes com valores legítimos

Melhorada função geradora de blocos aleatórios



Atacando a Layer 7: Vetores e Tools

Mitigação

Permissionamento granular das páginas

Filtrar POST onde não é necessário

Filtrar querystring onde não é necessário

ProxyCache

Utilizar proxies de cache (ex: Varnish) e não permitir reload

KeepAlive e TimeOuts

Acertar máximo do KeepAlive, TimeOut e KeepAliveTimeOut (Apache) e equivalentes em outros webservers

WebServer DoS Mitigation 101

Ataques de Denial-of-Service Super Eficientes. SEIDL, Jan

XVII Semana de Informática/2012 - Minas Gerais, Brasil

WebServer DoS Mitigation 101

Apache

LimitRequestFields, LimitRequestFieldSize, LimitRequestBody, LimitRequestLine, LimitXMLRequestBody, TimeOut, KeepAliveTimeOut, ListenBackLog, MaxRequestWorkers [core]

RequestReadTimeout [mod_reqtimeout]

Fonte: https://community.qualys.com/blogs/securitylabs/2011/11/02/how-to-protect-against-slow-http-attacks

WebServer DoS Mitigation 101

Nginx

client_max_body_size, client_body_buffer_size, client_header_buffer_size, large_client_header_buffers, client_body_timeout, client_header_timeout [core]

Modules: HttpLimitReqModule, HttpLimitZoneModule

Fonte: https://community.qualys.com/blogs/securitylabs/2011/11/02/how-to-protect-against-slow-http-attacks

WebServer DoS Mitigation 101

IIS 6 & 7

IIS 6: connectionTimeout, HeaderWaitTimeout, MaxConnections

IIS 7: <RequestLimits> maxAllowedContentLength, maxQueryString, maxUrl

<headerLimits>

<Limits>/<WebLimits> connectionTimeout, headerWaitTimeout, minBytesPerSecond

Fonte: https://community.qualys.com/blogs/securitylabs/2011/11/02/how-to-protect-against-slow-http-attacks

WebServer DoS Mitigation 101

USE UM WEB APPLICATION FIREWALL (WAF)

Modsecurity (Apache / Nginx)
http://www.modsecurity.org/

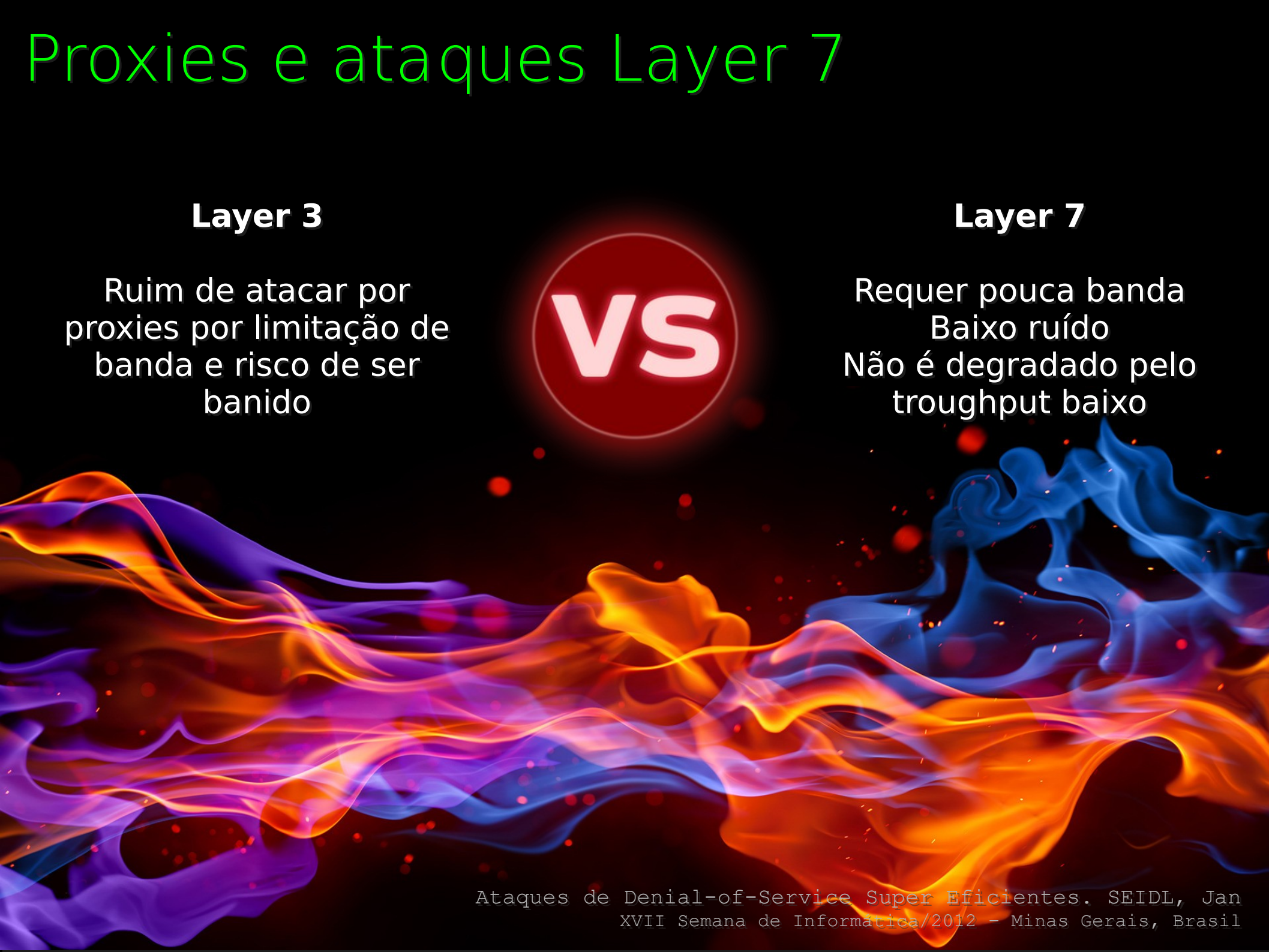
NAXSI (Nginx)
http://code.google.com/p/naxsi/

Proxies e ataques Layer 7

Proxies e ataques Layer 7

Layer 3

Ruim de atacar por proxies por limitação de banda e risco de ser banido



Layer 7

Requer pouca banda
Baixo ruído
Não é degradado pelo troughput baixo

Proxies e ataques Layer 7

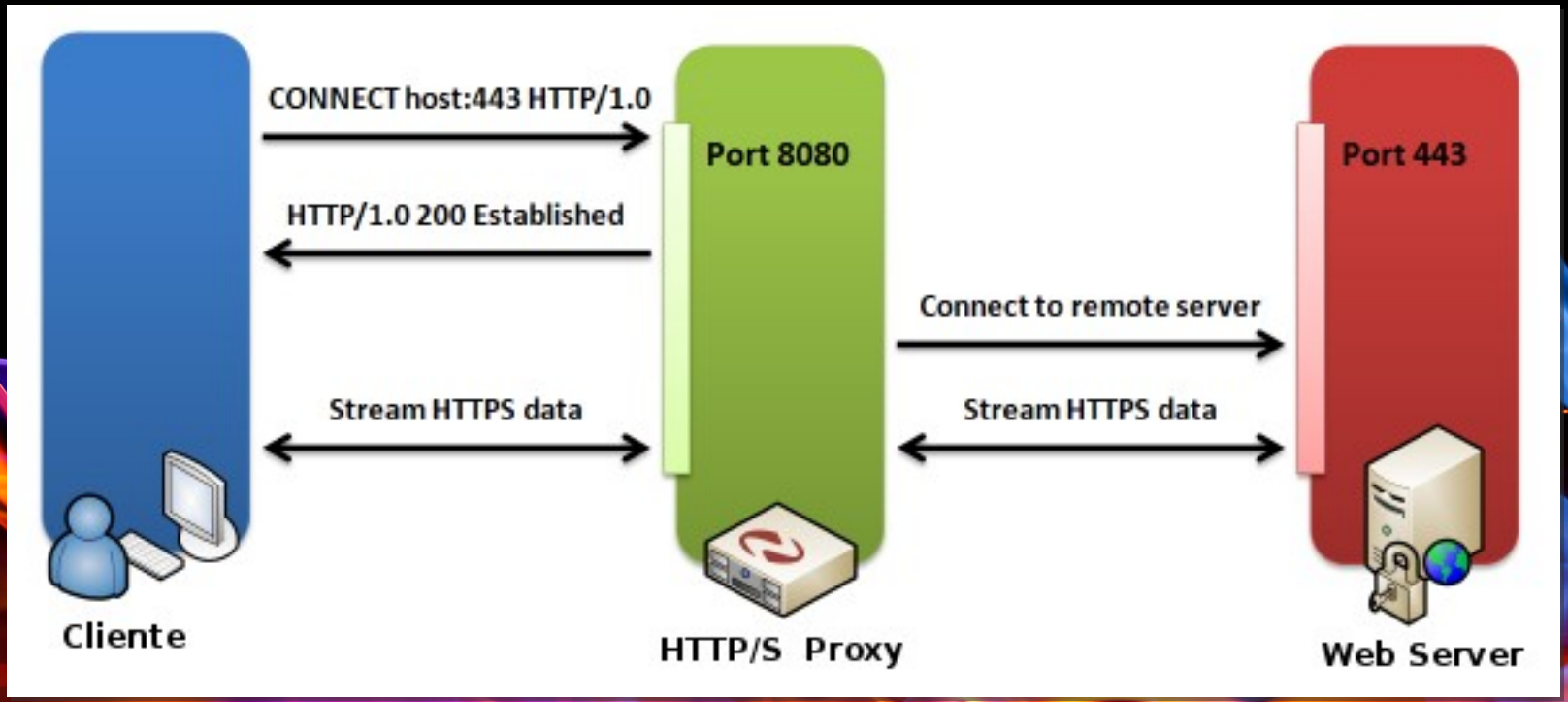
Por que empregar Proxies HTTP em ataques?

Resposta simples

- Ips diferentes
- Localização geográfica a escolha
- Podem fornecer alta anonimidade
- Largamente disponíveis na internet

Proxies e ataques Layer 7

Por que empregar Proxies HTTP em ataques?



Proxies e ataques Layer 7

Pivot de ataque por proxy

Ferramenta:

Socat: Multipurpose Relay
http://www.dest-unreach.org/socat/

Também com SSL:
HTTPS, IMAPS, POPS, LDAPS

Proxies e ataques Layer 7

Pivot de ataque por proxy: Regular Proxies

```
# socat TCP4-LISTEN:80
PROXY:<PROXY_IP>:<VICTIM_IP>:80,proxyport=<PROXY_PORT>

# echo “127.0.0.1 <VICTIM_HOST>” >> /etc/hosts

# ./goldeneye.py http://<VICTIM_HOST>/index.php -t 1000
-m get
```

Proxies e ataques Layer 7

Pivot de ataque por proxy: TOR

```
# socat TCP4-LISTEN:80,fork
SOCKS4A:localhost:<VICTIM_IP>:80,socksport=9052

# echo "127.0.0.1 <VICTIM_HOST>" >> /etc/hosts

# ./goldeneye.py http://<VICTIM_HOST>/index.php -t 1000
-m get
```

Proxies e ataques Layer 7

Bônus: Multi-TOR

A rede TOR permite que abram-se quantos túneis façam-se necessários.

```
tor --RunAsDaemon 1 --CookieAuthentication 0
--HashedControlPassword "pwd" --ControlPort 4444
--PidFile torN.pid --SocksPort 5090 --DataDirectory
data/torN
```

Ferramenta:

Multi-TOR
https://github.com/jseidl/Multi-TOR/

EX: ./multi-tor.sh 5 # Opens 5 TOR instances

Proxies e ataques Layer 7

Mitigando TOR com TORBlock

Bloqueando acesso via TOR

TORBlock: IPTables-based blocking

Ferramenta:

https://github.com/jseidl/torblock

Load Balancing Attacks

Meet Jericho

Load Balancing Attacks

Starring: HAProxy

“The Reliable, High Performance TCP/HTTP Load Balancer”

REQUEST → HAPROXY → { SERVER A, SERVER B, SERVER C }

Load Balancing Attacks

Anatomia do ataque 'load-balanced'

Atacante:

1. **Vários túneis socat para a vítima, cada um por um proxy diferente (regular ou TOR ou ambos)**

2. **Endereços das portas locais criadas com o socat no HAProxy**

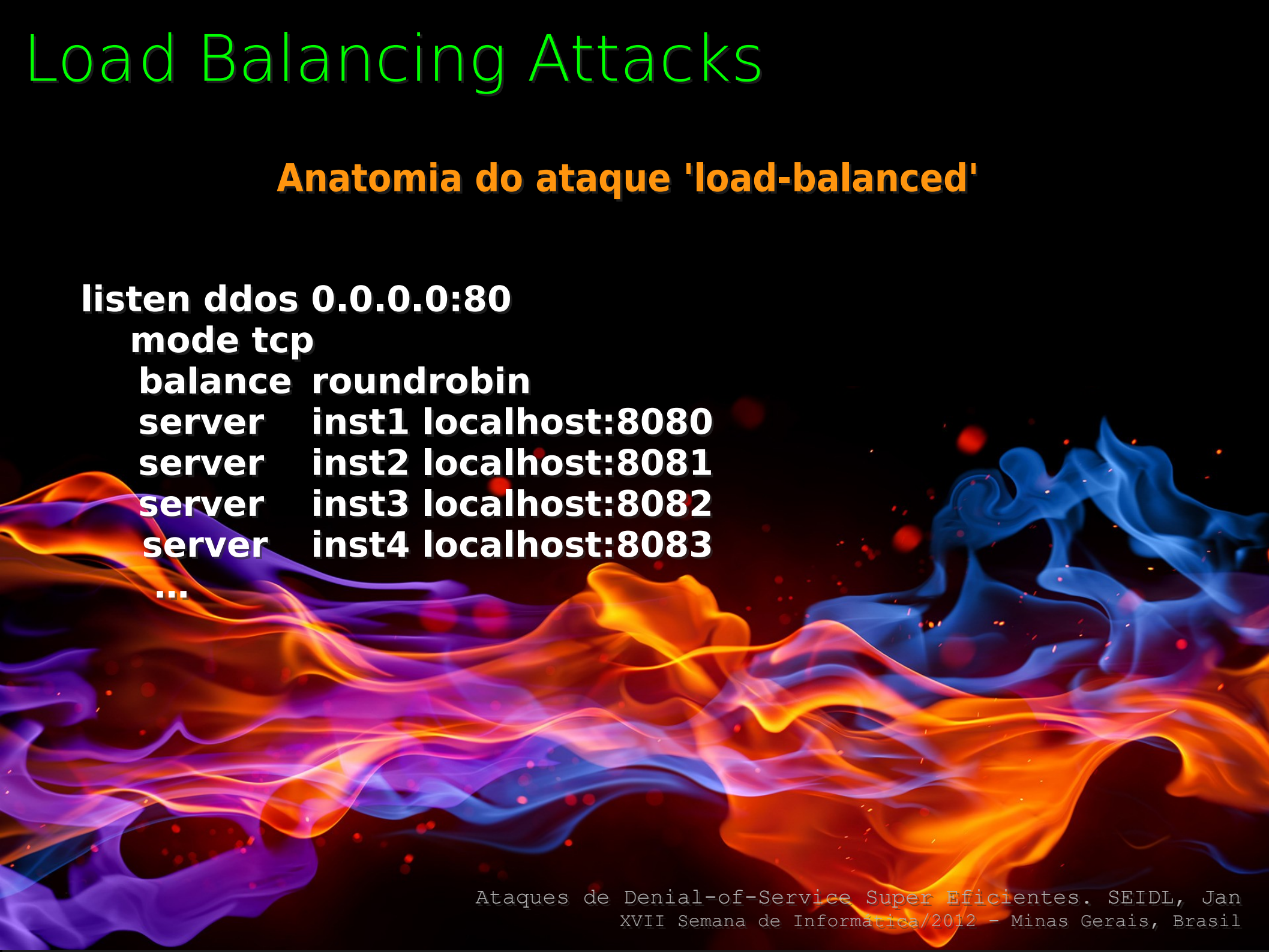
3. **Domínio da vítima no /etc/hosts**

4. **Ataque lancado normalmente pela tool desejada**

Load Balancing Attacks

Anatomia do ataque 'load-balanced'

```
listen ddos 0.0.0.0:80
    mode tcp
     balance  roundrobin
     server   inst1 localhost:8080
     server   inst2 localhost:8081
     server   inst3 localhost:8082
     server   inst4 localhost:8083
      ...
```

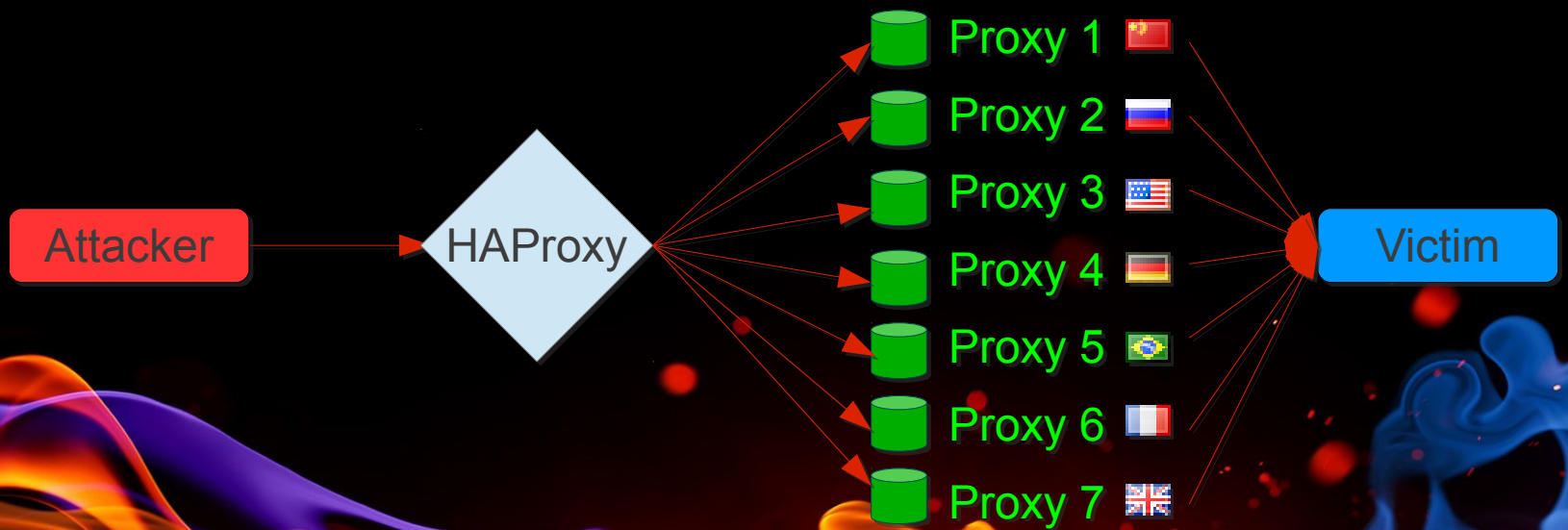


Load Balancing Attacks

Anatomia do ataque 'load-balanced'

Load Balancing Attacks

Anatomia do ataque 'load-balanced'



Load Balancing Attacks

Anatomia do ataque 'load-balanced'

Load Balancing Attacks

Perigos do ataque 'load-balanced'?

- Bypass connection-limiting
- DoS → DDoS
- Múltiplos Ips de origem
- Origem pode prover de vários países diferentes

Load Balancing Attacks

Perigos do ataque 'load-balanced'?

XSS D/DoS

E se uma falha em uma aplicação web pudesse tornar seus visitantes em ativistas D/DoS?

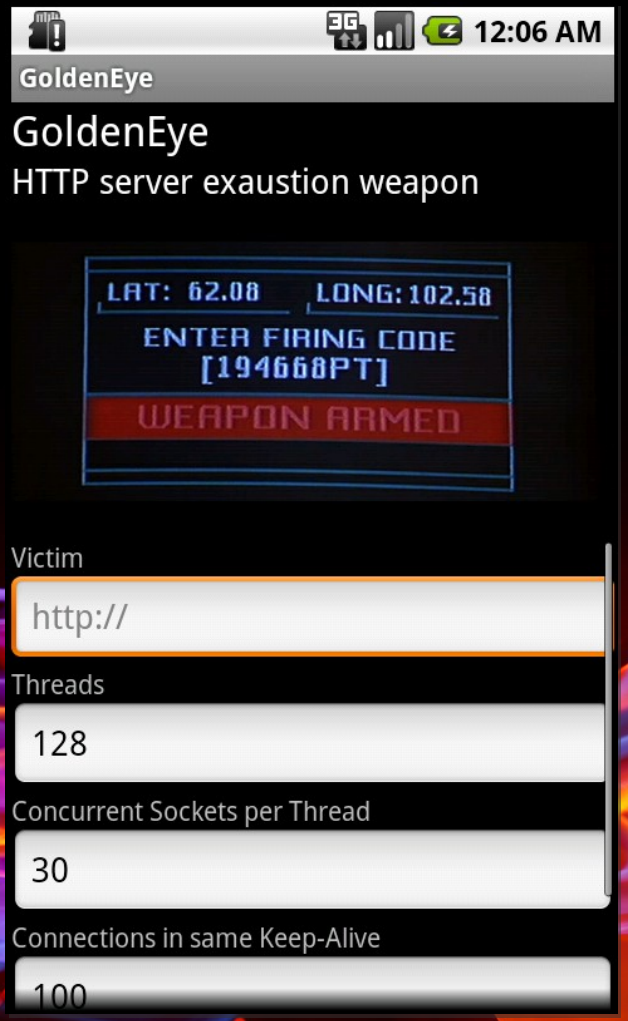
```
<script>
function DDoS() {
   a = new Date()
   unixepoch = a.getTime()

   elm = document.createElement("img")
   victimURL = "http://10.1.1.114/"
   elm.src = victimURL+"?"+unixepoch
}

setInterval("DDoS()",1);
</script>
```

Mobile-launched Denial-of-Service

PoC Tool: GoldenEye Mobile



Mobile-launched Denial-of-Service

Objetivo

Testar se os dispositivos móveis atuais poderiam conduzir sozinhos um ataque de DoS bem sucedido.

Testar se os equipamentos e configurações estão capazes de deter ataques de DoS oriundos de plataformas mobiles.

Mobile-launched Denial-of-Service

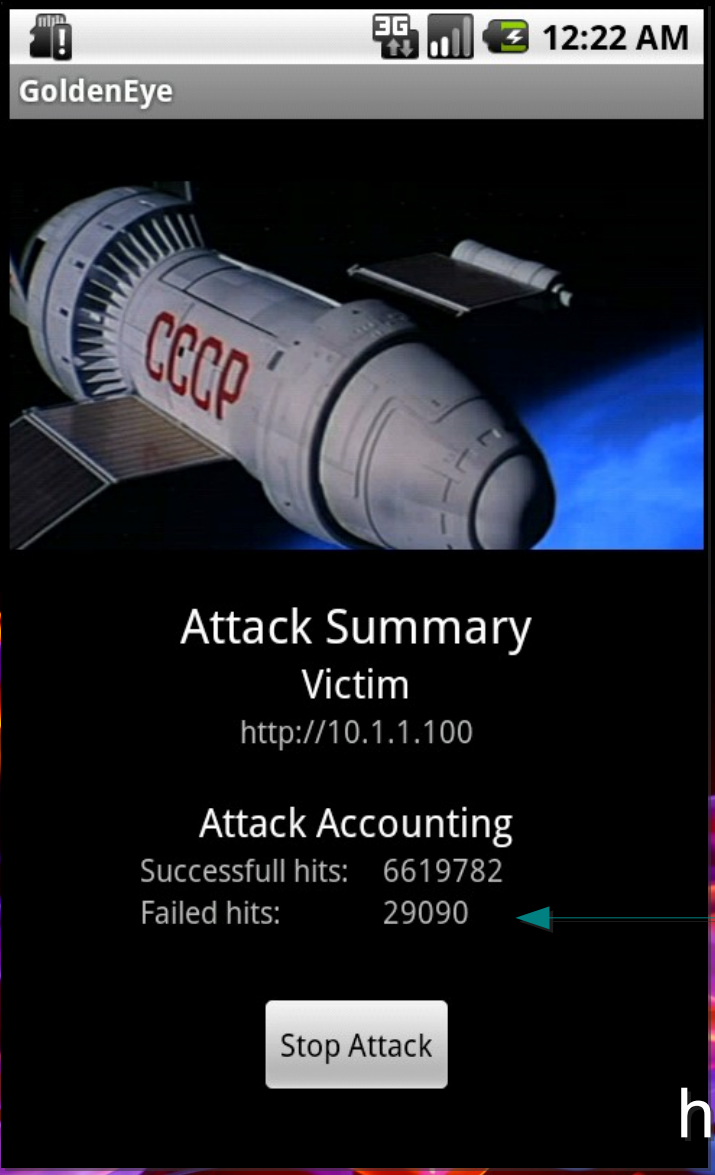
Android: Limitações

Máximo de 128 threads (Android 2.1)

Maximo de sockets por thread obtido: 30 (>30 too many open files)

Valor pode ser melhorado se o device for 'rooted' (sysctl) ?

Mobile-launched Denial-of-Service



Firepower

Teste de 5 minutos em um webserver Apache, configuração default, em máquina virtual Debian 6, também com configuração default.

```
CPU Usage: u5.85 s4.52 cu0 cs0 - 2.37% CPU load
```

Baixo fingerprint de CPU na vítima

Server sobrecarregado

https://github.com/jseidl/GoldenEye-Mobile

Mobile-launched Denial-of-Service

GoldenEye Mobile: Mitigação

GoldenEye Mobile usa o método HEAD para obter máxima velocidade.

Fácilmente bloqueável (Módulo: Mod_Rewrite)

```
RewriteEngine on
RewriteCond %{THE_REQUEST} !^(GET|POST)\ /.*\ HTTP/1\.1$
RewriteRule .* - [F]
```

mod_security

```
SecFilterSelective REQUEST_METHOD "!^(GET|POST)$" "deny,auditlog,status:405"
```

Demo: DoS Fun

GoldenEye Mobile DoS Android Tool Demo!

http://bit.ly/GoldenEyeMDOS

Dúvidas?

Obrigado!

- To Peace!

Obrigado!

Obrigado pela atenção!

jseidl@wroot.org

http://wroot.org

https://github.com/jseidl

http://www.slideshare.net/jseidl