Outpost24

The Technology Leader in Vulnerability Assessment and Management

Introduction to Sockstress

A TCP Socket Stress Testing Framework

Presented at the SEC-T Security Conference

Presented by:

Jack C. Louis – Senior Security Researcher, Outpost24

Creator of Sockstress

Robert E. Lee – CSO, Outpost24

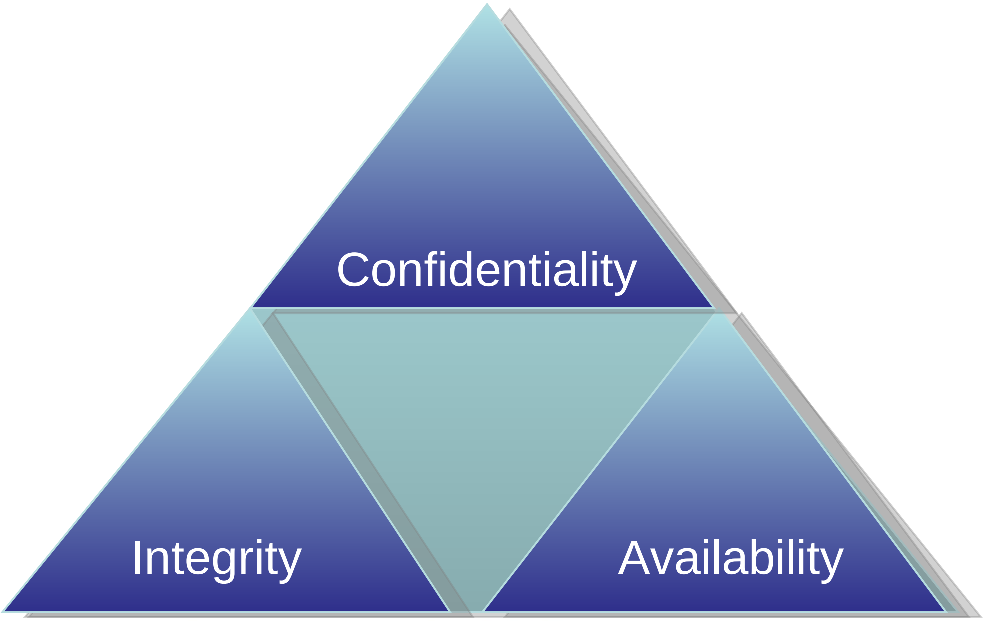
Goals of this talk

- Review TCP Sockets
- Discuss Historical TCP DoS Issues
- Reintroduce SYN Cookie Concept
- Present Sockstress

Problem Statement

Availability Critical to Function

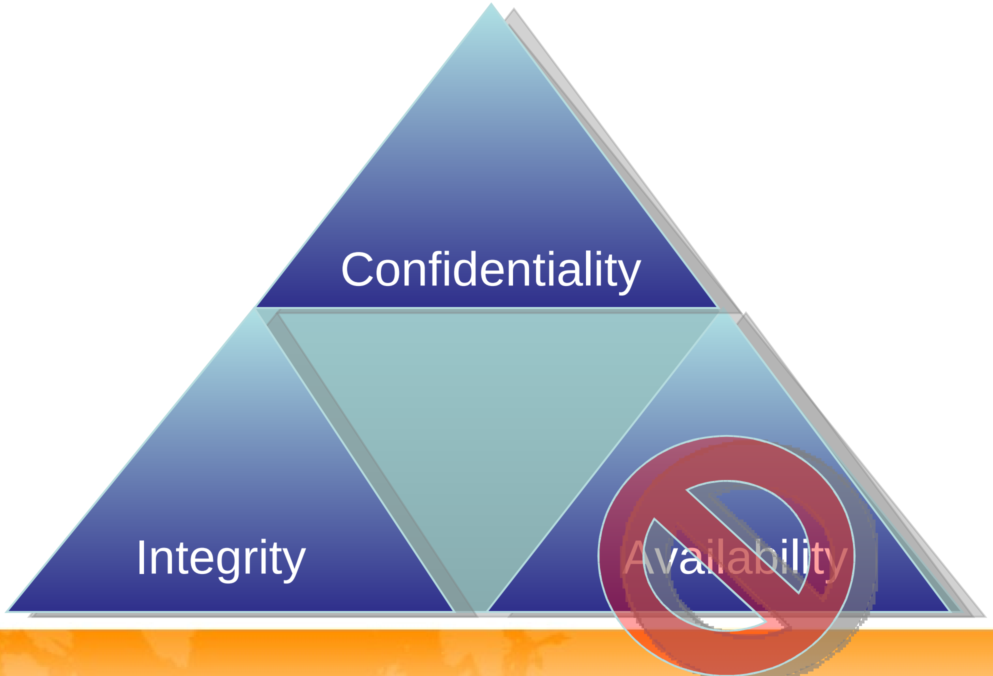
- Standard Security Triad – CIA



Problem Statement

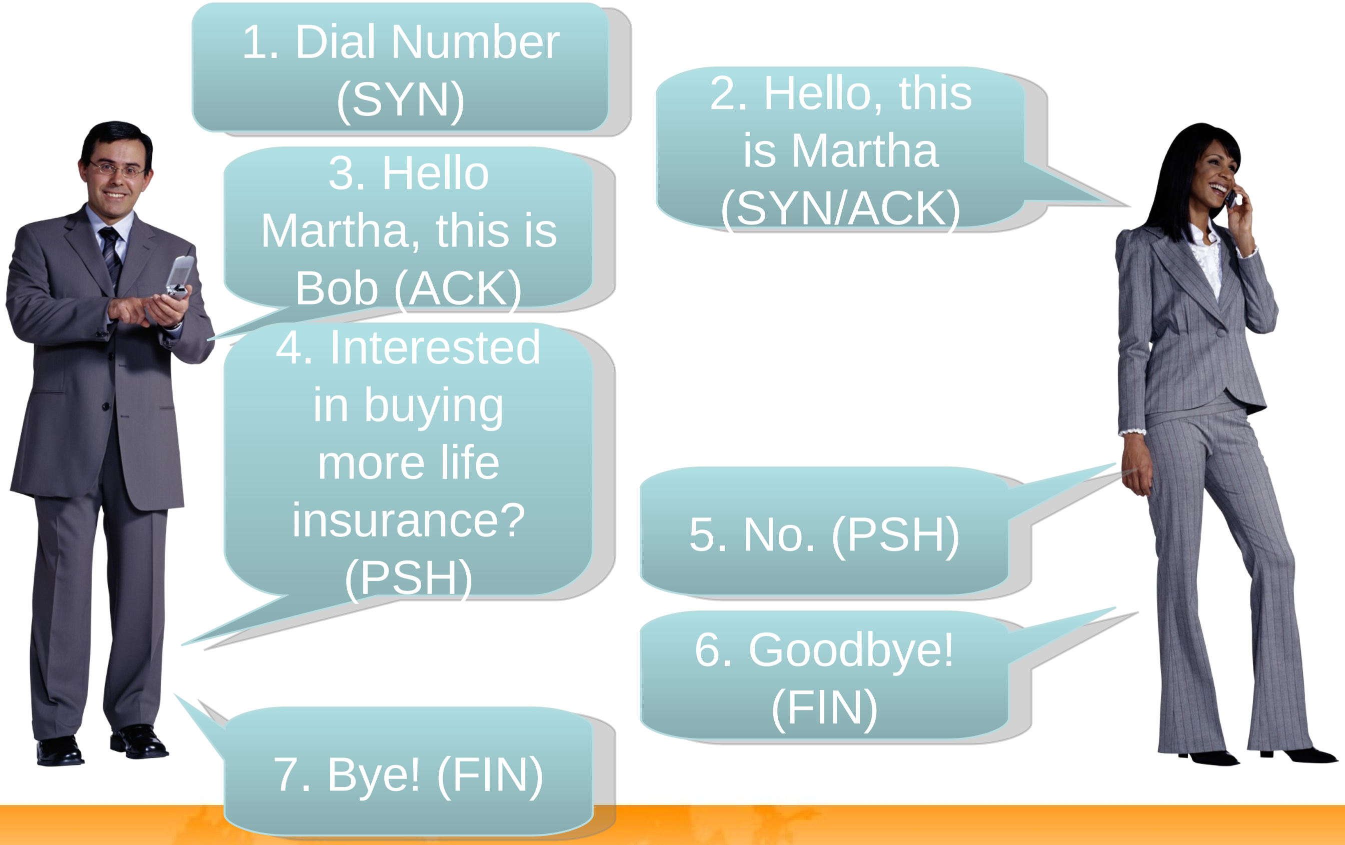
Availability Critical to Function

- Standard Security Triad – CIA
    - Without Availability, remaining security becomes less useful



TCP Connection Primer

Simplified example of a TCP Connection



1. Dial Number (SYN)

2. Hello, this is Martha (SYN/ACK)

3. Hello Martha, this is Bob (ACK)

4. Interested in buying more life insurance? (PSH)

5. No. (PSH)

6. Goodbye! (FIN)

7. Bye! (FIN)

States, Timers, & Counters

Every connection is tracked

- TCP connection states expire
  - Probe packets have max retries
- There are kernel defaults, but applications may also specify settings
- Applications can orphan connections

Legit User
192.168.1.1

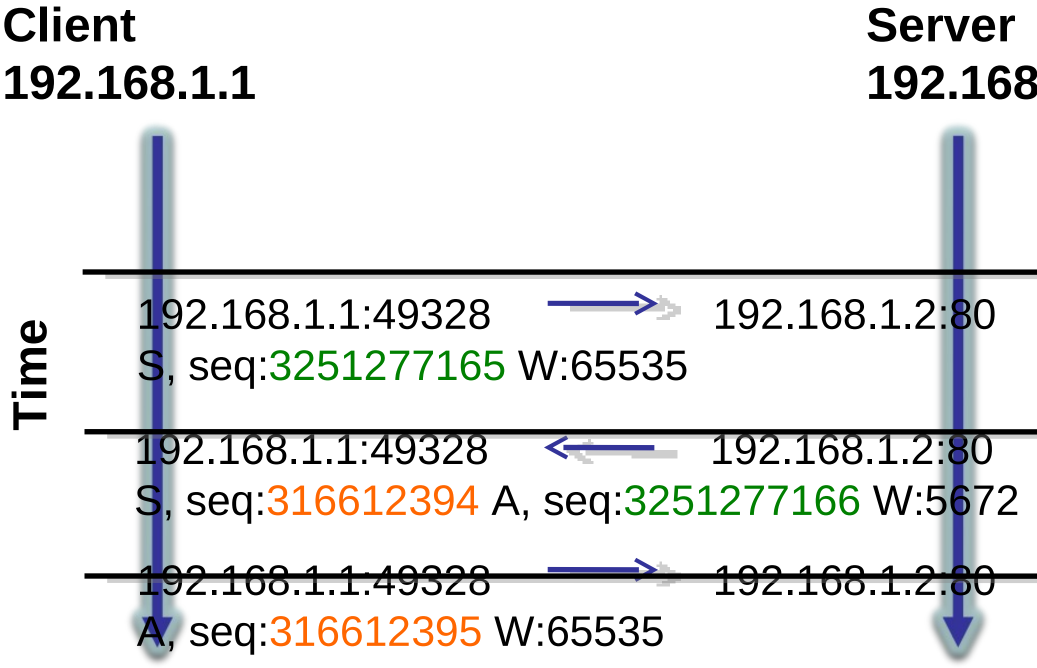
SYN

Server State Table

Local Address	Foreign Address	STATE	Timeout	Retries Left
192.168.1.2:80	192.168.1.1:49328	SYN_RCVD	75 Seconds	5

TCP Socket Connection

Introduction to the virtual circuit



Client
192.168.1.1

Server
192.168.1.2

Time

192.168.1.1:49328 → 192.168.1.2:80
S, seq:3251277165 W:65535

192.168.1.1:49328 ← 192.168.1.2:80
S, seq:316612394 A, seq:3251277166 W:5672

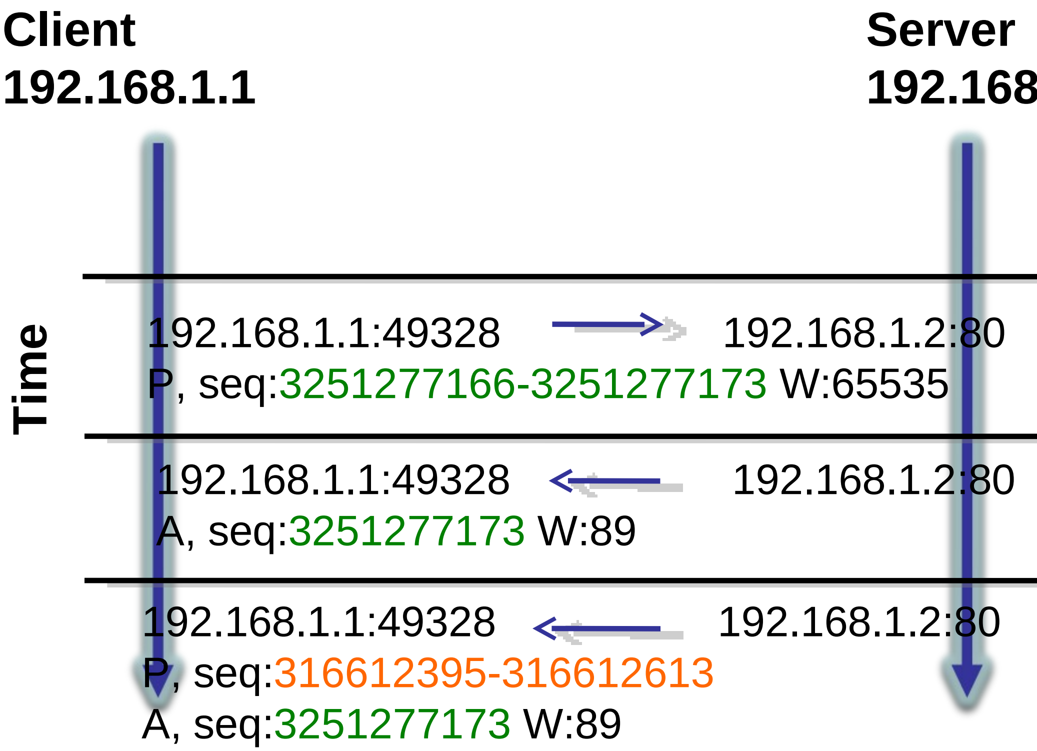
192.168.1.1:49328 → 192.168.1.2:80
A, seq:316612395 W:65535

Server State Table

Local Address	Foreign Address	STATE
*:80	*:*	LISTEN
192.168.1.2:80	192.168.1.1: 49328	SYN_RCVD
192.168.1.2:80	192.168.1.1: 49328	ACK_WAIT
192.168.1.2:80	192.168.1.1: 49328	ESTABLISHED

TCP Socket Connection

Introduction to the virtual circuit – Continued



Client
192.168.1.1

Server
192.168.1.2

Time

192.168.1.1:49328 → 192.168.1.2:80
P, seq:3251277166-3251277173 W:65535

192.168.1.1:49328 ← 192.168.1.2:80
A, seq:3251277173 W:89

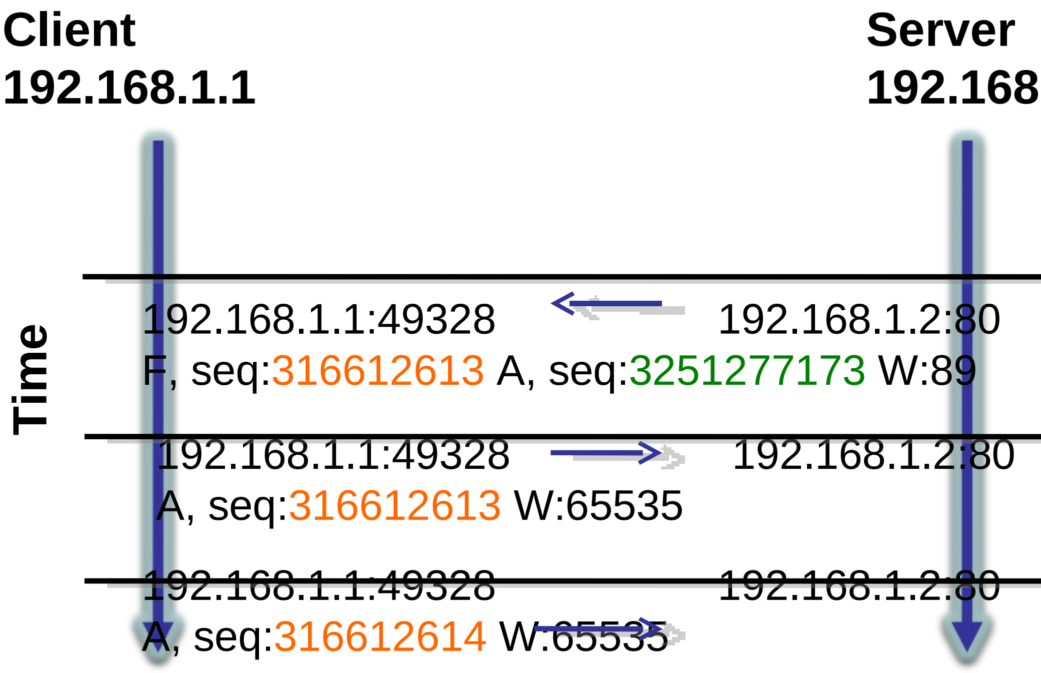
192.168.1.1:49328 ← 192.168.1.2:80
P, seq:316612395-316612613
A, seq:3251277173 W:89

Server State Table

Local Address	Foreign Address	STATE
192.168.1.2:80	192.168.1.1: 49328	ESTABLISHED
192.168.1.2:80	192.168.1.1: 49328	ESTABLISHED
192.168.1.2:80	192.168.1.1: 49328	ESTABLISHED

TCP Socket Connection

Introduction to the virtual circuit – Continued



192.168.1.1:49328 192.168.1.2:80
F, seq:316612613 A, seq:3251277173 W:89

192.168.1.1:49328 192.168.1.2:80
A, seq:316612613 W:65535

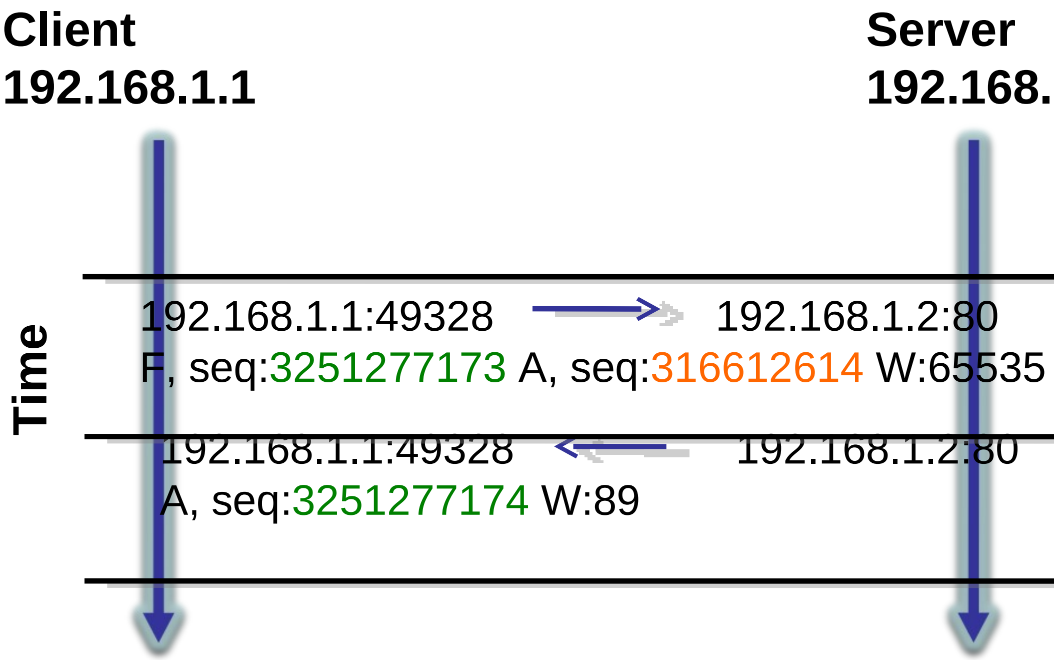
192.168.1.1:49328 192.168.1.2:80
A, seq:316612614 W:65535

Server State Table

Local Address	Foreign Address	STATE
192.168.1.2:80	192.168.1.1: 49328	FIN_WAIT_1
192.168.1.2:80	192.168.1.1: 49328	FIN_WAIT_1
192.168.1.2:80	192.168.1.1: 49328	FIN_WAIT_2

TCP Socket Connection

Introduction to the virtual circuit – Continued



192.168.1.1:49328 → 192.168.1.2:80
F, seq:3251277173 A, seq:316612614 W:65535

192.168.1.1:49328 ← 192.168.1.2:80
A, seq:3251277174 W:89

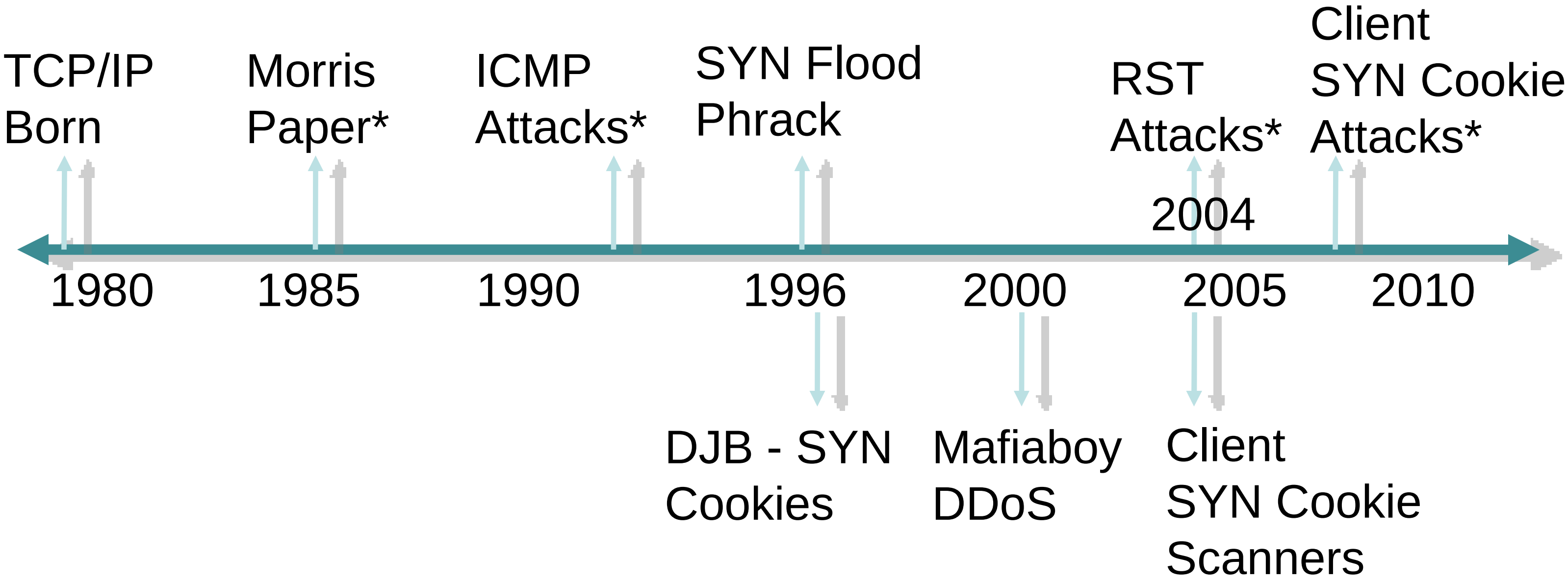
Server State Table

Local Address	Foreign Address	STATE
192.168.1.2:80	192.168.1.1:49328	CLOSE_WAIT_1
192.168.1.2:80	192.168.1.1:49328	CLOSED
*:80	*	LISTEN

DoS Timeline for **TCP**

TCP has been around since 1979

- In it’s history, only 4 major DoS attack types for the general protocol.
  - SYN Flood, ICMP, RST, Client SYN



TCP/IP Born

Morris Paper*

ICMP Attacks*

SYN Flood Phrack

RST Attacks*

Client SYN Cookie Attacks*

2004

1980 1985 1990 1996 2000 2005 2010

DJB - SYN Cookies

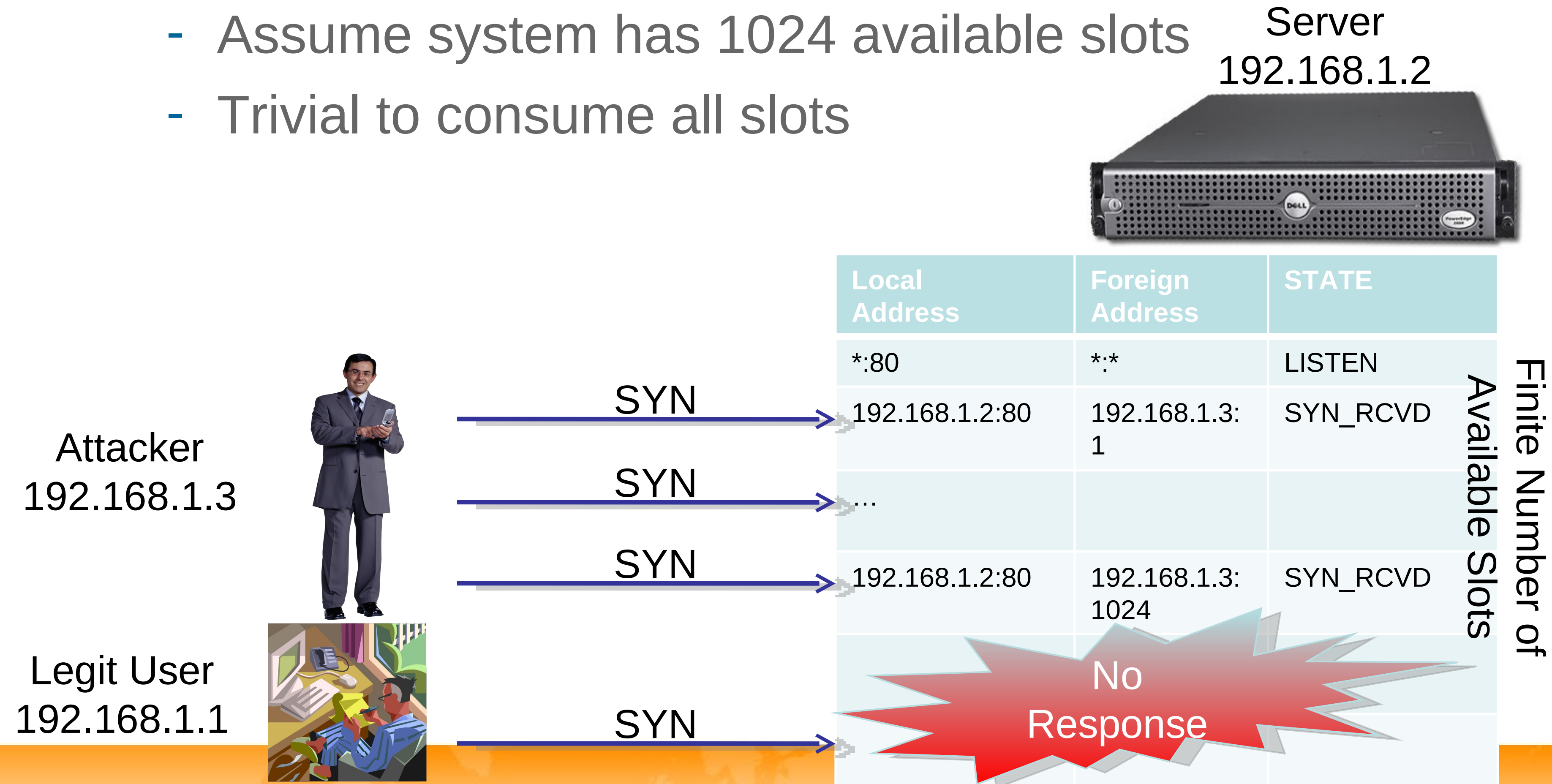
Mafiaboy DDoS

Client SYN Cookie Scanners

SYN Flood

Every connection attempt must be accounted for

- Assume system has 1024 available slots
- Trivial to consume all slots

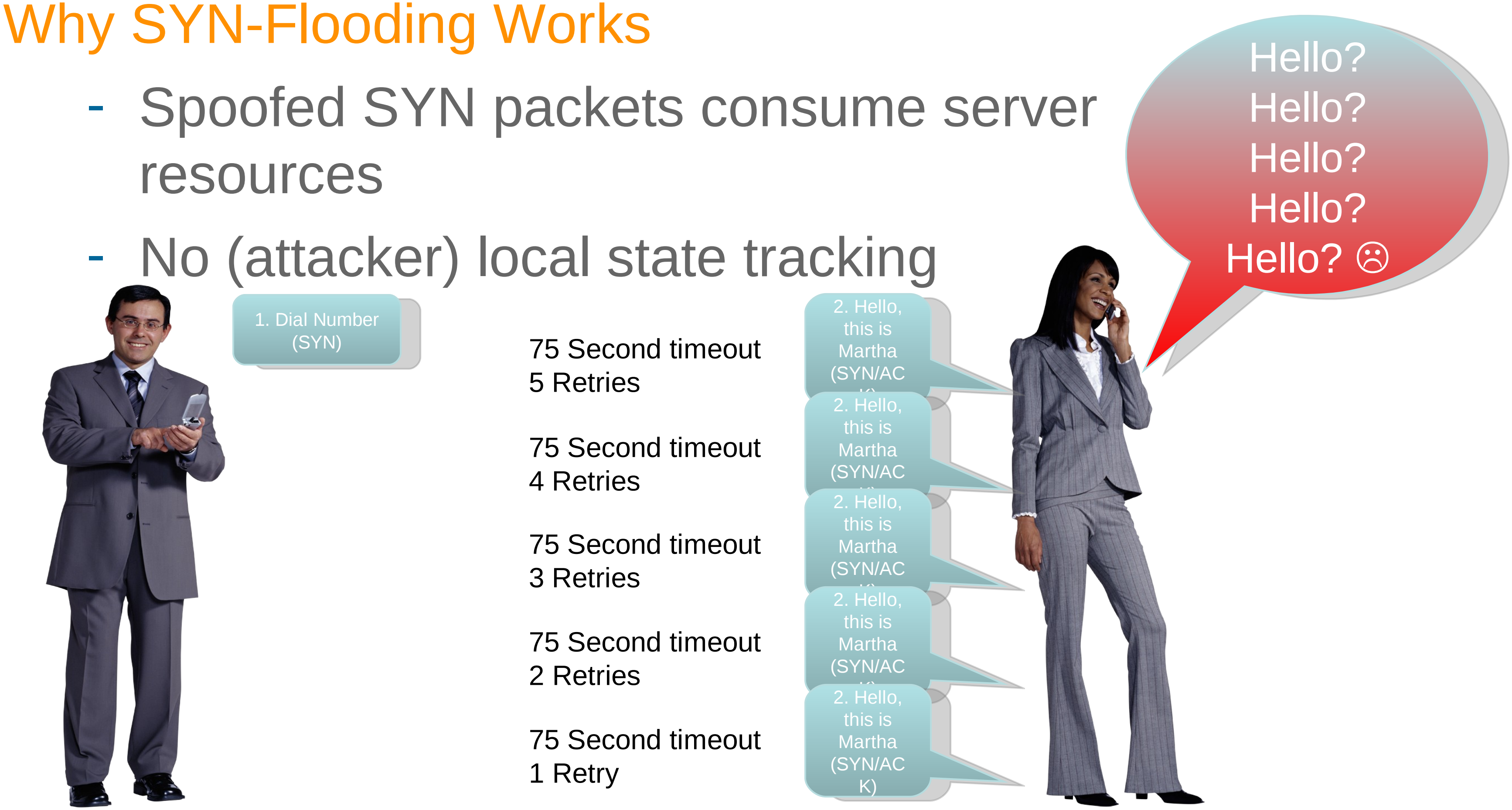


Local Address	Foreign Address	STATE
*:80	*:*	LISTEN
192.168.1.2:80	192.168.1.3: 1	SYN_RCVD
…		
192.168.1.2:80	192.168.1.3: 1024	SYN_RCVD

SYN Flood

Why SYN-Flooding Works

- Spoofed SYN packets consume server resources
- No (attacker) local state tracking



1. Dial Number (SYN)

75 Second timeout
5 Retries

75 Second timeout
4 Retries

75 Second timeout
3 Retries

75 Second timeout
2 Retries

75 Second timeout
1 Retry

2. Hello, this is Martha (SYN/AC

2. Hello, this is Martha (SYN/AC

2. Hello, this is Martha (SYN/AC

2. Hello, this is Martha (SYN/AC

2. Hello, this is Martha (SYN/ACK)

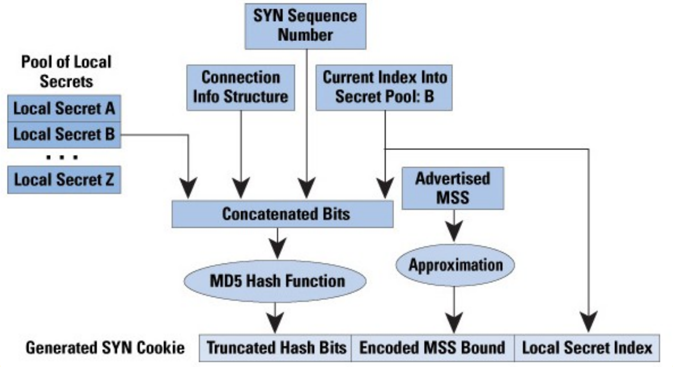
Hello?
Hello?
Hello?
Hello?
Hello? ☹

SYN Cookies

How to combat SYN Flooding

- SYN Cookies defer TCP Connection State Tracking until after 3-way handshake
- SYN Cookie is sent by Server as Initial Sequence Number
  - Cookie is hashed meta-data representing the connection details

SYN Cookie Hash

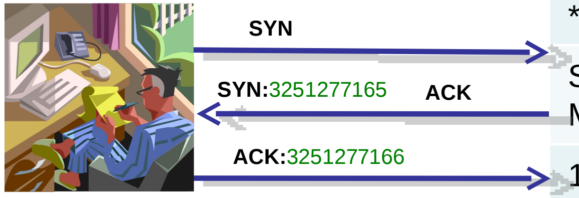


SYN Cookies

How to combat SYN Flooding – Continued

- When ACK of ISN received, server compares (response - 1) to hash list
  - If match found, state is ESTABLISHED
  - Otherwise, rejected

Legit User
192.168.1.1



Local Address	Foreign Address	STATE
*:80	*:*	LISTEN
SYN Cookie Magic		
192.168.1.2:80	192.168.1.1: 49328	ESTABLISHED

SYN Cookie Hash Table

3251277165	Meta-Data

SYN Cookies

How to combat SYN Flooding – Continued

- Requiring valid cookie response:
    - Ensures attacker must see SYN/ACK responses (is a “legitimate IP address”)
        - Requires attacker to consume resources to account for state
    - Reduced resource load on server
        - Frees connection slots for other legit users

Full Connection Flood

Why Full Connection Flooding isn’t more popular

- A full connection requires attacker to consume state tracking resources too

Oh no! No more outgoing lines. ☹
1. Dial Number (SYN)
2. Hello, this is Martha (SYN/ACK)
3. Hello Martha, this is Bob (ACK)
1. Dial Number (SYN)
2. Hello, this is Martha (SYN/ACK)
3. Hello Martha, this is Bob (ACK)
1. Dial Number (SYN)
2. Hello, this is Martha (SYN/ACK)
3. Hello Martha, this is Bob (ACK)
1. Dial Number (SYN)
2. Hello, this is Martha (SYN/ACK)
1. Dial Number (SYN)
2. Hello, this is Martha (SYN/ACK)
3. Hello Martha, this is Bob (ACK)

Defeating SYN Cookies

Fight Fire with Fire

- To defeat Server side SYN Cookies...
  - Employ Client side SYN Cookies

- Start with a random 32-bit number
- XOR this number against Client side of a connection attempt (192.168.1.3:51242)
- Use output as ISN for SYN packets

Defeating SYN Cookies

Fight Fire with Fire – Continued

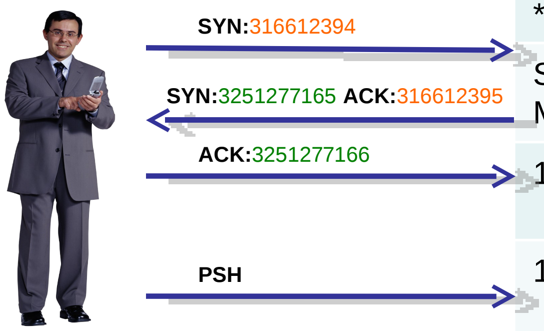
- When Client receives SYN/ACK's
  - (Sequence Number - 1) XOR'd with 32-bit number reveals the client sending IP and port
- Client can now complete a full 3 way handshake without ever tracking anything in a table.
  - Client can also transmit data on this connection

Defeating SYN Cookies

Fight Fire with Fire – Continued

- No need on Client side to even keep a hash table. XOR is reversible.

Attacker
192.168.1.3



Local Address	Foreign Address	STATE
*:80	*:*	LISTEN
SYN Cookie Magic		
192.168.1.2:80	192.168.1.3: 51235	ESTABLISHED
192.168.1.2:80	192.168.1.3: 51235	ESTABLISHED

SYN Cookie Hash Table

3251277165	Meta-Data

Sockstress Attacks

To be seen and experienced live at the show…

- We are still working with vendors, so we must limit the details of what Sockstress is Attacking
  - We will share more background information at the talk
  - We will also demonstrate the attacks live

Outpost24

One Step Ahead!