Computer Network Security

COMP 178 | Spring 2025 | University of the Pacific | Jeff Shafer

Bonus Topic:

Network Address Translation (NAT) & Slipstream Attacks

Upcoming Assignments

- **Lab 6 – Post Exploitation:** Due Feb 26th
- **Lab 7 – Password Testing:** Due March 5th
- **Video Presentation**
  - Due February 25th – TODAY!
  - Upload video and slides to separate Canvas assignments
- **Video Presentation Peer Reviews – 3 each**
  - Canvas will auto-assign on March 2nd (look in the same assignment where you uploaded the video)
  - Due March 9th

NAT and Slipstream Attacks

- Today's agenda
  - Network Address Translation (NAT)
  - Slipstream Attack v1
  - Slipstream Attack v2

Review: Private IPv4 Addressing

- Not routable on public internet
  - No chance of conflict with a valid public IP
- Why do I want private addresses?
  - Not every printer / phone / IOT device / etc. needs to be publicly accessible from the Internet
  - Useful for local collections of computers not connected to Internet

Name	IP address range	Number of IPs
10.0.0.0/8	10.0.0.0 – 10.255.255.255	16,777,216
172.16.0.0/12	172.16.0.0 – 172.31.255.255	1,048,576
196.168.0.0/16	192.168.0.0 – 192.168.255.255	65,536

Review: TCP and UDP

- Two common protocols nested inside IP packets

TCP

- Reliability guaranteed
- Connection-based
  - Stream of data between two endpoints
  - Must explicitly open and close

UDP

- Delivery not guaranteed
- No connections
  - Each packet is independent (like IP)

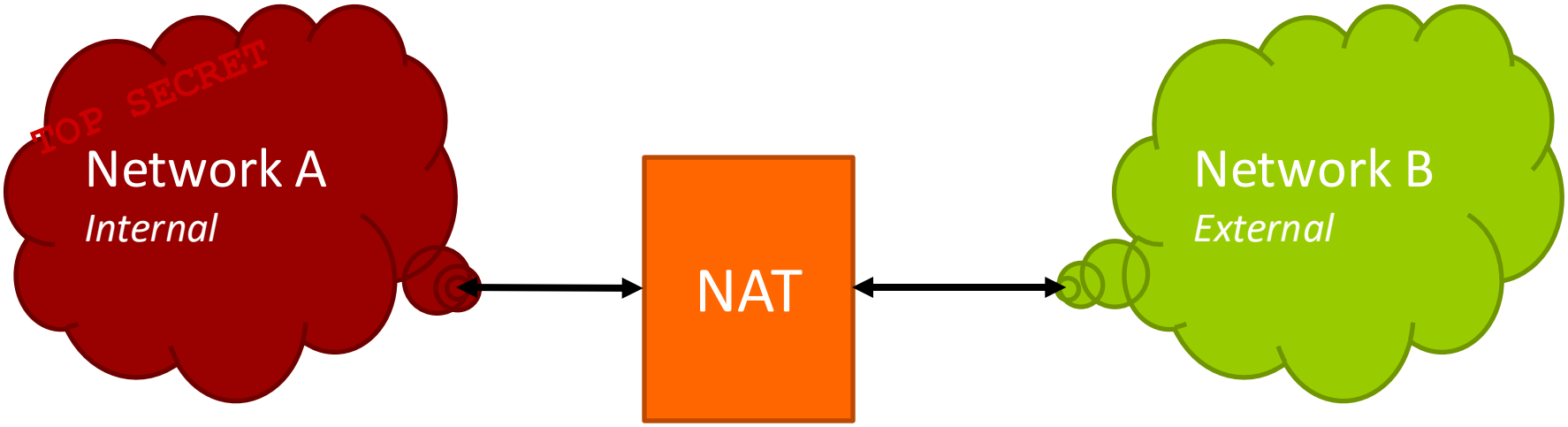
- Each protocol uses port numbers to distinguish between independent data streams

Network Address Translation

- Translate / route packets between one IP address space and another
  - Commonly translates from private IP range to public IP range (but the concept can be generalized to two public address ranges)
- Accomplished by modifying packet header
  - Source address
  - Destination address
  - IP port number
  - IP / TCP / UDP checksums

Not every NAT technique modifies every field!

Network Address Translation



- Network A
  - Multiple computers trying to access network B
  - Don't want to reveal network A's structure to network B
- Network B
  - Traffic from network A appears with addresses in Network B's space
  - May be mapped as single or multiple addresses

Why Use Address Translation?

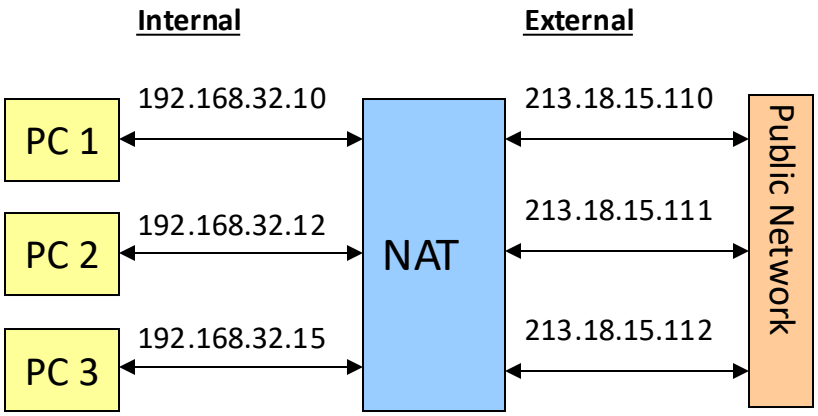
- Allows multiple hosts on private network to access public network through a single address
  - Overcomes policy problems
    (e.g. buying extra IPs from your ISP costs $$)
  - Overcomes IPv4 address shortages
- Disguises internal network structure
  - All requests appear to originate from NAT unit
  - Increases “security”
- Allows you to use entire `10.x.x.x` private address space and remap to smaller public address range
  - Very convenient for clean network topology and simplified router forwarding tables

Types of Translation

- Terms are used interchangeably
- **N**etwork **A**ddress **T**ranslation (**NAT**)
  - Translates only the address fields, not ports
  - Every machine on network A gets a unique address on network B
- **P**ort **A**ddress **T**ranslation (**PAT**)
  - Translates address and port numbers
  - Allows multiple machines on network A to share single IP address on network B
    - All requests appear to come from PAT unit

Network Address Translation Types

- One-to-One Mapping
  - Every internal IP gets a different external IP
- Static
  - Internal IP always mapped to same External IP
- Dynamic / Pooled
  - Internal IP is mapped to random external IP

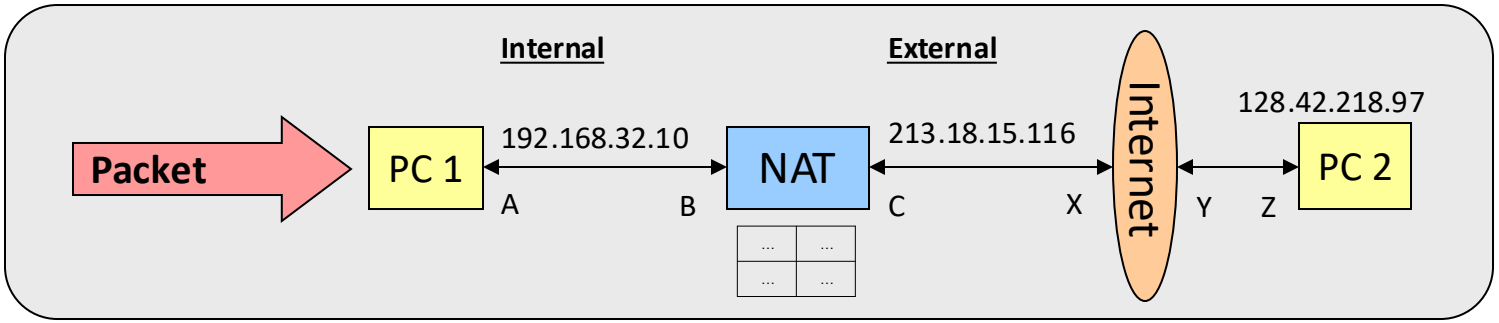


NAT Mapping Table: Static or Dynamic

Internal IP	External IP
192.168.32.10	213.18.15.116
192.168.32.12	213.18.15.112
192.168.32.15	213.18.15.125
…	…

Not shown in Table: MAC Addresses!

NAT Mechanics – Outbound Packet



Ethernet Header			IP Header					Data	
Dst MAC	Src MAC	...	...	IP Csum	Src IP	Dst IP	...	Payload	CRC

Before NAT (internal network)

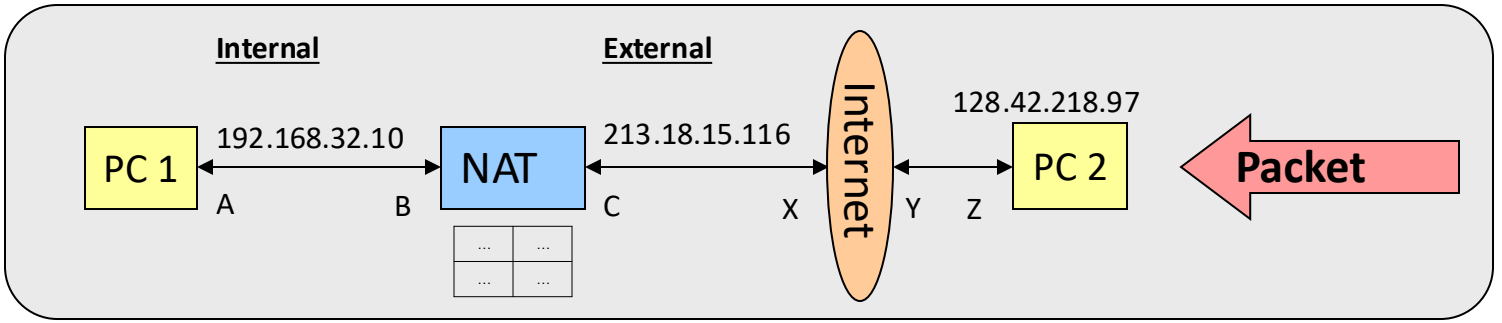
B	*A*	*...*	*...*	*IP Csum*	*PC 1*	*PC 2*	*...*	*Payload*	*CRC*

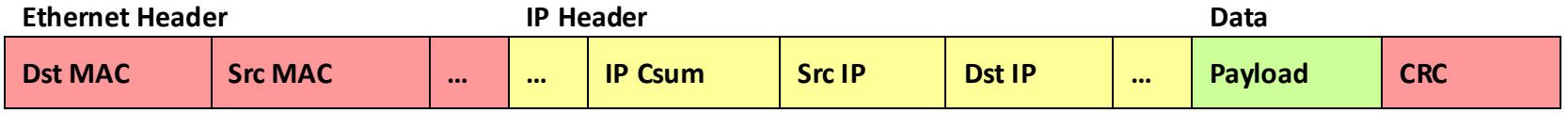
After NAT (external network)

X	*C*	*...*	*...*	*IP Csum*	*NAT*	*PC 2*	*...*	*Payload*	*CRC*

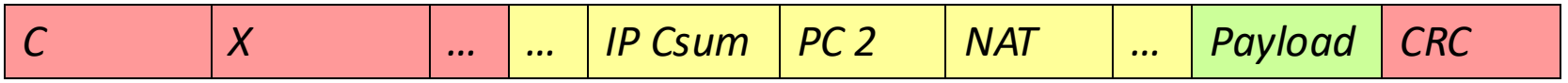
- Save internal IP and MAC to mapping table
- Replace source IP and MAC with NAT unit
- Recalculate checksums (Ethernet CRC, IP header, TCP/UDP/... headers)

NAT Mechanics – Inbound Packet

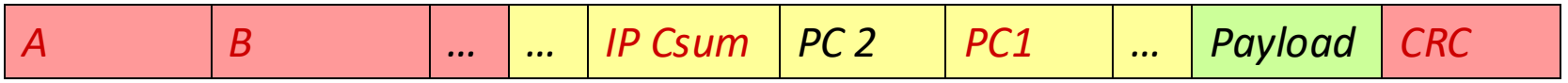




Before NAT (external network)

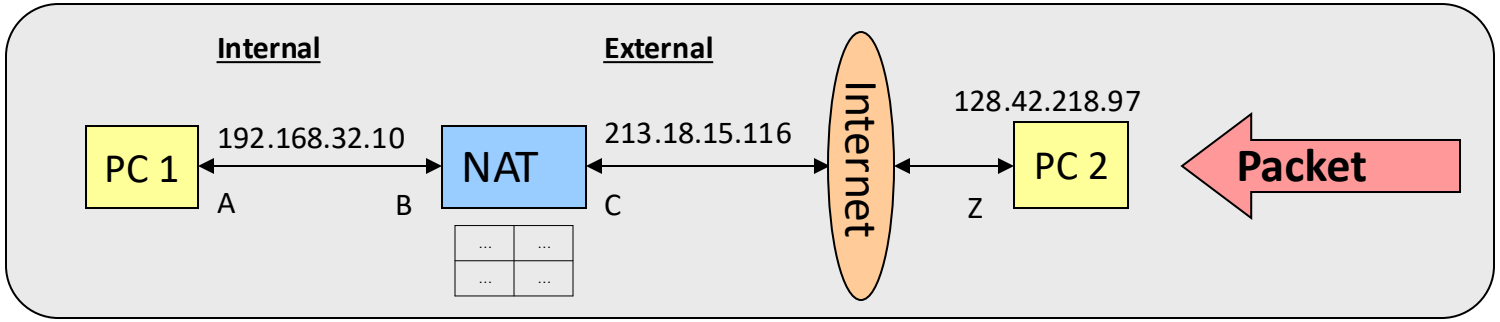


After NAT (internal network)



- Lookup Dst IP in mapping table. Only forward if match found
- Replace Dst IP and MAC with private address
- Update checksums (CRC, IP, TCP/UDP/...)

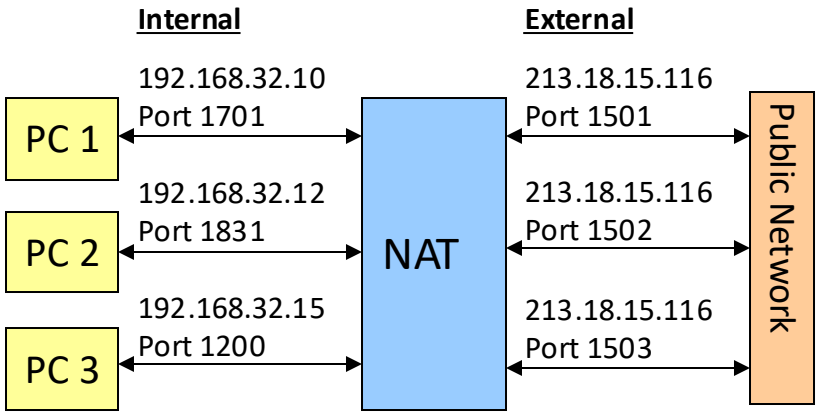
NAT Mechanics – Inbound Packet



- What happens if a router sends a packet to the NAT unit, but no valid mapping exists for the destination IP?
  - **Packet is dropped**

Port Address Translation

- IP Overloading
  - Many internal IPs are mapped to one (or a few) external IPs
  - TCP/UDP port number is also changed and used to identify unique connections between internal and external hosts
  - Typically dynamic



NAT Mapping Table

Internal IP	Internal Port	External IP	External Port
192.168.32.10	1701	213.18.15.116	1501
192.168.32.12	1831	213.18.15.116	1502
192.168.32.15	1200	213.18.15.116	1503
…	…	…	…

Not shown in Table: MAC Addresses!

Clearing Mappings

- When should a mapping be removed from a NAT?
  - Static NAT – Never?
  - Dynamic NAT – Only if the host is idle for a long time?
- When should a mapping be removed from a PAT?
  - TCP – Close of connection or reasonable timeout
    - Connection is framed by SYN and FIN packets
  - UDP – Unable to determine close of “connection”, so must use reasonable timeout instead

NAT/PAT – Protocol Challenges

- **PAT Fails**: Protocols that require incoming connections
  - Example: FTP Active Mode
    - Client sends request
    - Server attempts to open new connection back to client to send data
    - No entry in PAT table so connection is rejected
  - Example: SIP / RTP (VOIP telecommunication)
- **NAT / PAT Fails**: Protocols that carry IP address / port values in their payload
  - Example: IPsec (and other tunneling / VPN protocols)
    - NAT changes src/dst addresses in header but is unable to fix encrypted payload. Packet fails security check and is discarded because receiver detects (correctly) that the packet was altered in transit
- **NAT / PAT Fails**: Protocols that use checksums which include IP addresses
  - NAT only knows how to recalculate checksums for IP/TCP/UDP packets, not any new protocol that might be developed

Application-Level Gateway (ALG)

- Technique to avoid breaking common protocols
- NAT device runs multiple ALGs
  - Each ALG looks for a different protocol
  - Rewrites packet payload to fix problems
- Common ALG modules
  - FTP, SIP, H.323, RTSP, IPSec, etc…
- Not future proof
  - Each ALG is a fix for a specific protocol
  - Need to upgrade NAT software as new applications are developed

Severs and PAT

- Is there a simple way to enable servers to function behind a PAT?
- Administrator can insert static mappings into mapping tables
  - e.g. All incoming TCP requests on port 80 should always be forwarded to IP A.B.C.D, port 80 (enables a web server)
- Must be configured in advance
- Doesn't scale well
  - What if I have two web servers behind my PAT?
  - What if I don't know the incoming port #?
- Can be automated via Universal Plug and Play (UPnP) Internet Gateway Device (IGD) Protocol
  - This is designed for home use, not a corporate datacenter

Severs and NAT

- Do I need to do anything to get my servers behind NAT to work?
  - No – IP address mapping is already one-to-one
  - A static mapping would be helpful for the clients…

NAT and Security

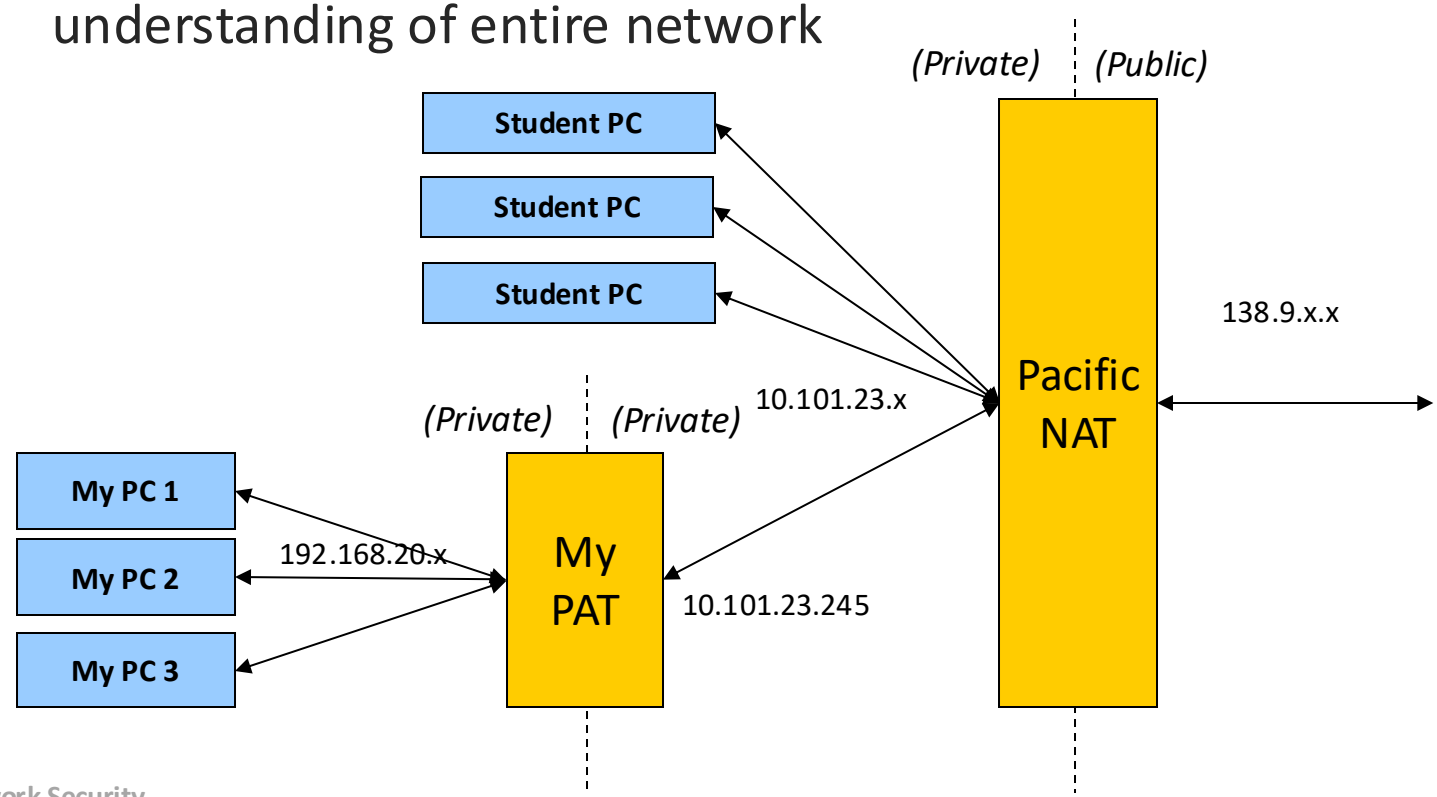
- NAT is often advertised as being essential for security
- Security through obscurity?
  - *"If evil hacker on public network can't see me, I must be secure!"*
  - Computers on private network using PAT are hidden
    - Protects against worms scanning for exploits as long as there are no static mappings allowing outside access
    - If your parents have a simple PAT in front of their unpatched Windows box, they're protected against some worms

NAT and Security

- Provides no protection against whole classes of malware
  - A security flaw in your PDF viewer can still be exploited by a bad download
  - The user can still do dangerous / stupid things ("Click on Angelina_Jolie.exe for free pictures!")
- Limited protection on larger networks
  - Servers must be publicly accessible to perform their function (via fixed port or IP mapping)
  - If your IIS webserver or Linux server with remote SSH is unpatched, it is still vulnerable to worms
  - Once compromised, this machine provides entry vector to reach internal network, which may be completely unprotected!
- Don't let your guard down - Security in depth

Nesting IP Ranges via NAT

- Allowed to have multiple levels of NAT
  - Each level performs translation independently without any understanding of entire network



NAT Slipstreaming

NAT Slipstreaming

- **Method to bypass NATs and firewalls to reach devices on internal network**
- NAT Slipstreaming **v1**
  - Vuln can open external access to any port **on <u>your</u> device behind your NAT**
  - By Samy Kamkar
  - Disclosed Oct 31 2020
- NAT Slipstreaming **v2**
  - Vuln can open external access to any port **on <u>any</u> device behind your NAT**
  - By Ben Seri, Gregory Vishnipolsky (w/Samy Kamkar)
  - Disclosed Jan 26 2021

https://github.com/samyk/slipstream

NAT Slipstreaming v2.0

- General scenario
  - Internal network full of vulnerable devices
    - Industrial controllers? Security cameras? IOT? Printers?
    - *Devices never intended to be on the public Internet*
    - *Devices with default logins*
    - *Devices with unpatched software*
  - Devices “protected” by a NAT/firewall that only allows outbound access
    - **Perimeter security is the only real security present**
  - Slipstream attack tricks NAT into adding forwarding entries, making these internal devices accessible from public Internet

NAT Slipstreaming v2.0 Demo

https://www.youtube.com/watch?v=ZAEDu3kLR1o

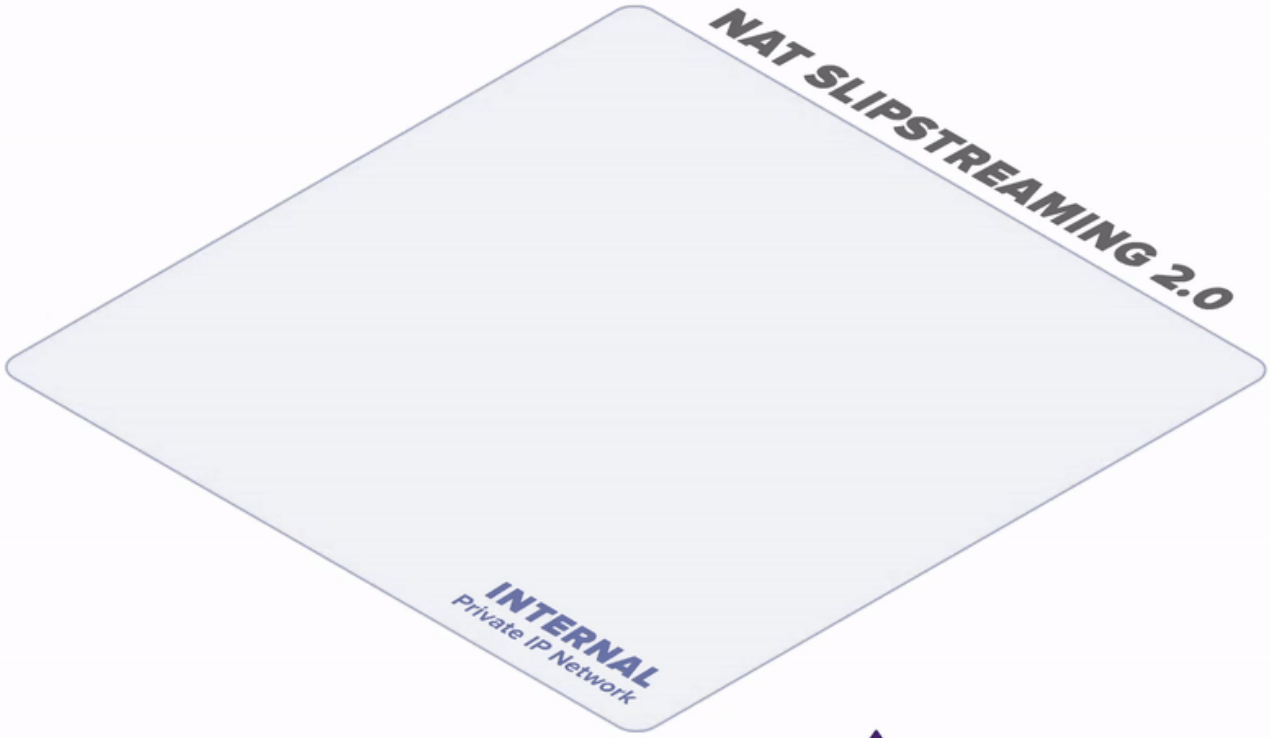
NAT Slipstreaming v2.0

- Demo of implications of slipstreaming attack in an "OT" (operational technology, i.e. industrial) network
- **See Also**: Similar demo of same attack in an enterprise network (targeting a printer and security camera)
  - https://www.youtube.com/watch?v=M-6ppoYDEV4
- How does it *work?*

https://www.armis.com/resources/iot-security-blog/nat-slipstreaming-v2-0-new-attack-variant-can-expose-all-internal-network-devices-to-the-internet/

NAT Slipstreaming v2.0

1. Attacker sends malicious link to www.igotcha.com
2. User clicks on www.igotcha.com
3. Malicious website runs code in browser
4. Secondary web requests fool the NAT to expose multiple private IP addresses to the Internet
5. Attacker now has access to all devices
6. Specific device is identified for attack

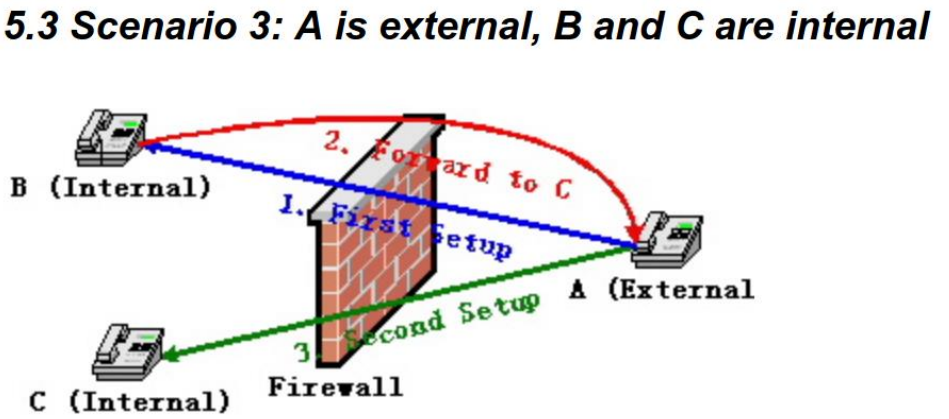


ARMIS | @SamyKamkar

https://www.armis.com/resources/iot-security-blog/nat-slipstreaming-v2-0-new-attack-variant-can-expose-all-internal-network-devices-to-the-internet/

H.323 ALG

- H.323 is a protocol used by VoIP (telephone)
- Pinhole in NAT (mapping to internal `IP:port`) must be created by Application Level Gateway (ALG) so that phone is reachable by external callers
  - H.323 port: **1720**
- Key "feature" (for slipstream attack) is that H.323 supports **call forwarding** and thus a good ALG should too



H.323 ALG

- "My Phone": 10.1.0.3, port 52286
- "Other phone": 10.0.0.69, port 1720
- "Forwarded-To Phone": 10.1.08, port 80 (the target we want to be publicly accessible)

```
▸ Internet Protocol Version 4, Src: 10.1.0.3, Dst: 10.0.0.69
▸ Transmission Control Protocol, Src Port: 52286, Dst Port: 1720,
▸ TPKT, Version: 3, Length: 71
▸ Q.931
▾ H.225.0 CS
  ▾ H323-UserInformation
    ▾ h323-uu-pdu
      ▾ h323-message-body: facility (6)
        ▾ facility
            protocolIdentifier: 0.0.8.2250.0.5 (Version 5)
          ▾ alternativeAddress: ipAddress (0)
            ▾ ipAddress
                ip: 10.1.0.8
                port: 80
          ▸ reason: callForwarded (1)
          ▸ callIdentifier
          ▸ h245Address: ipAddress (0)
            fastStart: 0 items
            1... .... multipleCalls: True
            1... .... maintainConnection: True
        0... .... h245Tunnelling: False
```

NAT Slipstreaming v2.0

- **Really Clever Bit**
  - A web browser doesn't natively speak H.323 – it isn't a VOIP phone. How can the attacker fake a H.323 conversation?
  - The ALG doesn't track entire conversations (too memory intensive, too many TCP packets)
    - Just looks for a single TCP packet going to port 1720 where the contents match H.323 fields - **stateless**
  - Web browser (running attacker-controlled JavaScript) sends *large* HTTP Fetch request to attacker server, port 1720
  - Uses **padding bytes** so that attacker-controlled bytes fit perfectly into a TCP packet by themselves – **NAT won't see the difference!**
  - Might take multiple attempts but attacker can loop and try again with different amount of padding

Remediation

- Remediated with web browser patches:
  - Slipstreaming v1: CVE-2020-16022 (Chrome) and other web browsers
  - Slipstreaming v2: CVE-2020-16043 (Chrome), CVE-2021-23961 (Firefox), CVE 2021-1799 (Safari)
- Browsers (Chrome et. al.) now block these ports from all HTTP/HTTPS/FTP communication

69	**TFTP**	1723	**H.323**
137	**NetBIOS**	5060	**SIP**
161	**SNMP**	5061	**SIP**
1719	**H.323**	6566	**SANE**
1720	**H.323**	10080	

Remediation

- Unresolved questions:
  - **Can a pentester exploit this by non-web browser means?** (Other methods of running arbitrary code on client inside network)
  - **Can NAT/router/firewall vendors tighten up their ALGs?** (Without breaking the purpose of the ALG?)