BluePrint: Automatic Malware Signature Generation for Internet Scanning

Kevin Stevens, Mert Erdemir, Hang Zhang, Taesoo Kim, Paul Pearce

Talk Overview

First system able to generate **Internet-scanning signatures** for server-like malware

Presentation structure:

→ Introduce an **example** piece of malware

→ Explain how to **scan** for it

→ Explain how our **system** works

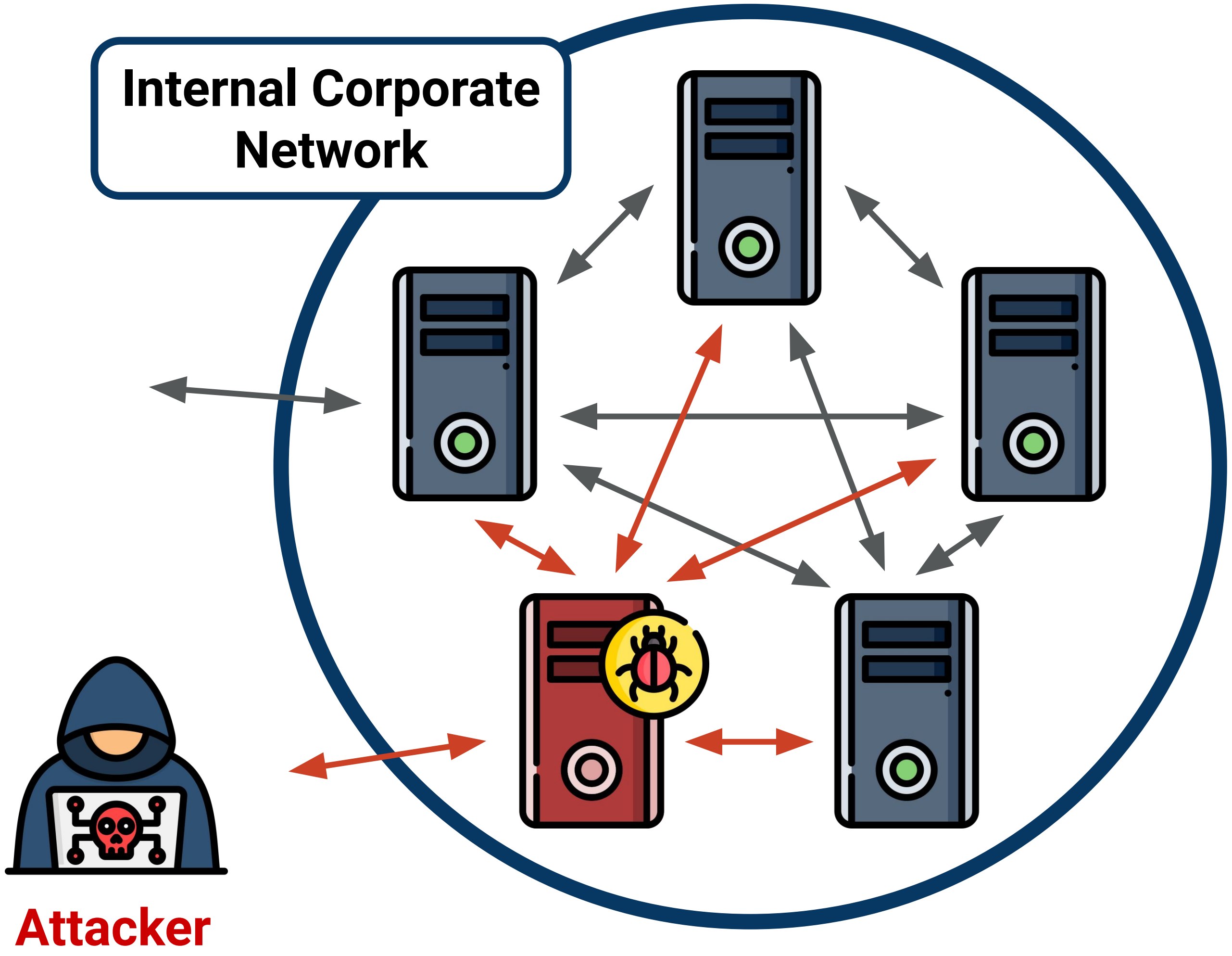
→ Explain **evaluation** results

Introduction to *BankShot*

“Proxy malware” from 2016 or 2017

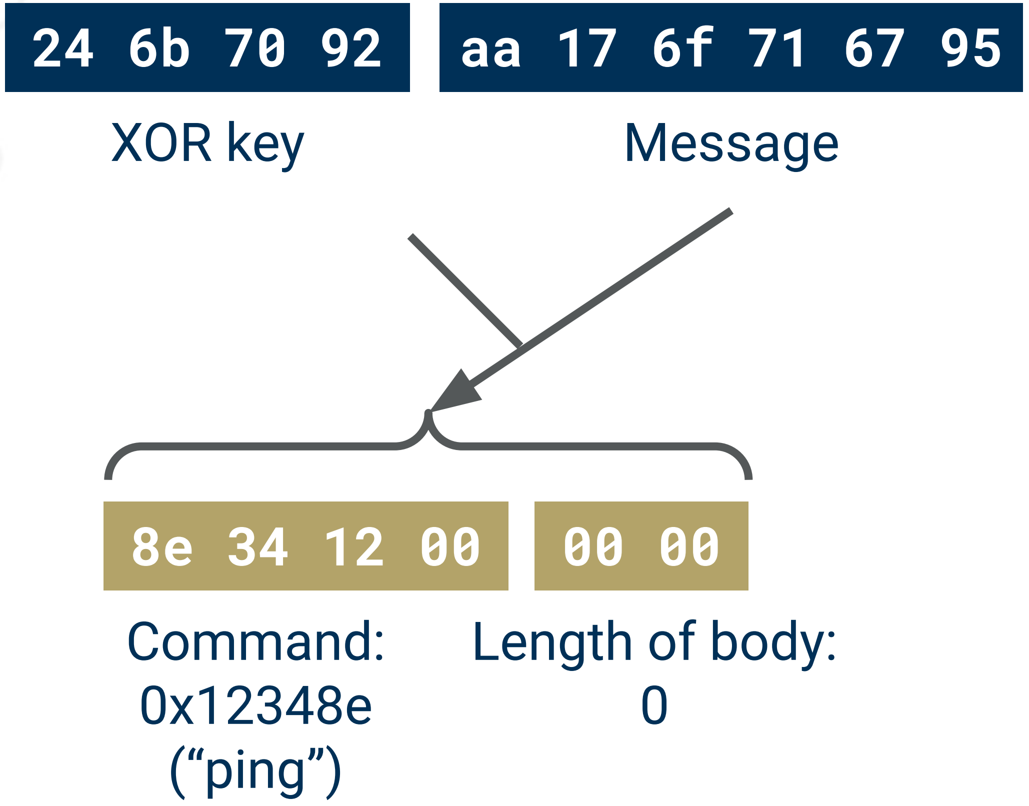
Attributed by US government to **North Korea**

Likely for network reconnaissance and data exfiltration



BankShot Ping Command

Challenge (Remote → Malware)



`24 6b 70 92` `aa 17 6f 71 67 95`

XOR key

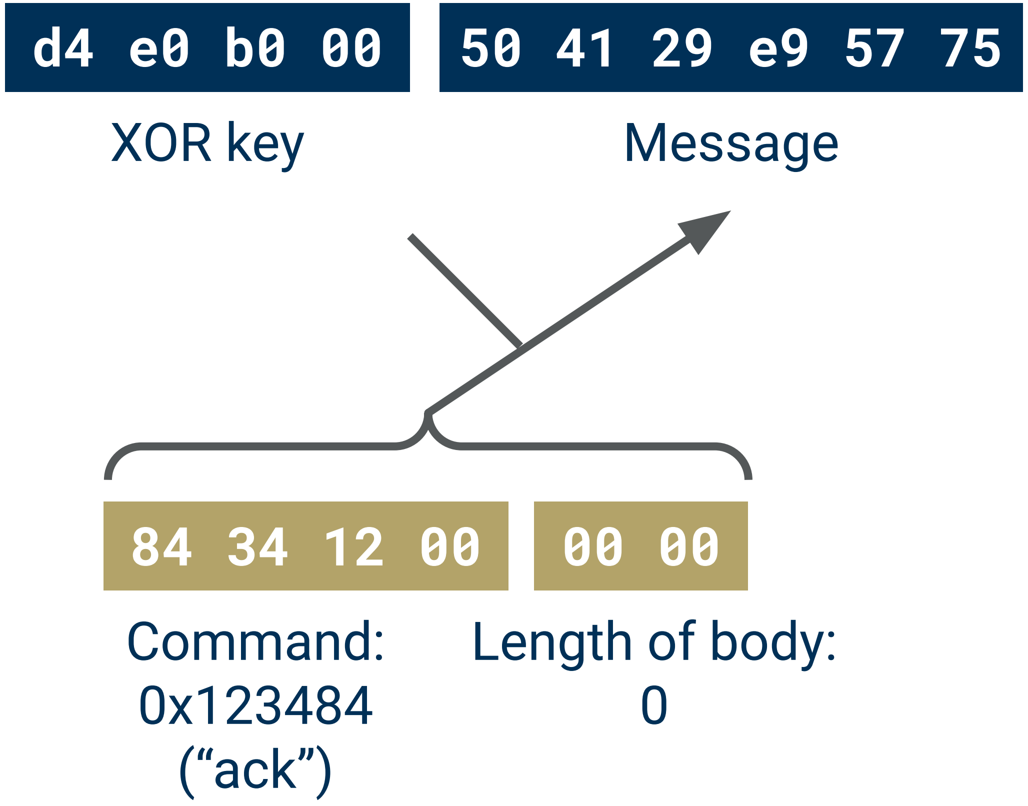
Message

`8e 34 12 00` `00 00`

Command:
0x12348e
("ping")

Length of body:
0

Response (Remote ← Malware)



`d4 e0 b0 00` `50 41 29 e9 57 75`

XOR key

Message

`84 34 12 00` `00 00`

Command:
0x123484
("ack")

Length of body:
0

How To Find Malware like *BankShot* in the Real World

Ability to perform population studies and identify real-world compromises is **crucial**.

Approaches:

Endpoint Security Systems

- + Full visibility on each system
- – Requires large install base, not available to most researchers

Infiltrating C&C Infrastructure

- + Comprehensive
- – High manual effort, slow
- – Often not possible
- – Legal/ethical concerns

How To Find Malware like *BankShot* in the Real World

Ability to perform population studies and identify real-world compromises is **crucial**.

Endpoint Sec

- + Full visibility c
- – Requires larg available to m

Internet Scanning

- + High coverage (*e.g.*, all IPv4)
- + Can be done legally/ethically
- + Requires only a fast, cooperative ISP
- – Only possible for some malware
- – ~~High manual reverse engineering and scanner setup effort~~ **(until now!)**

Infrastructure

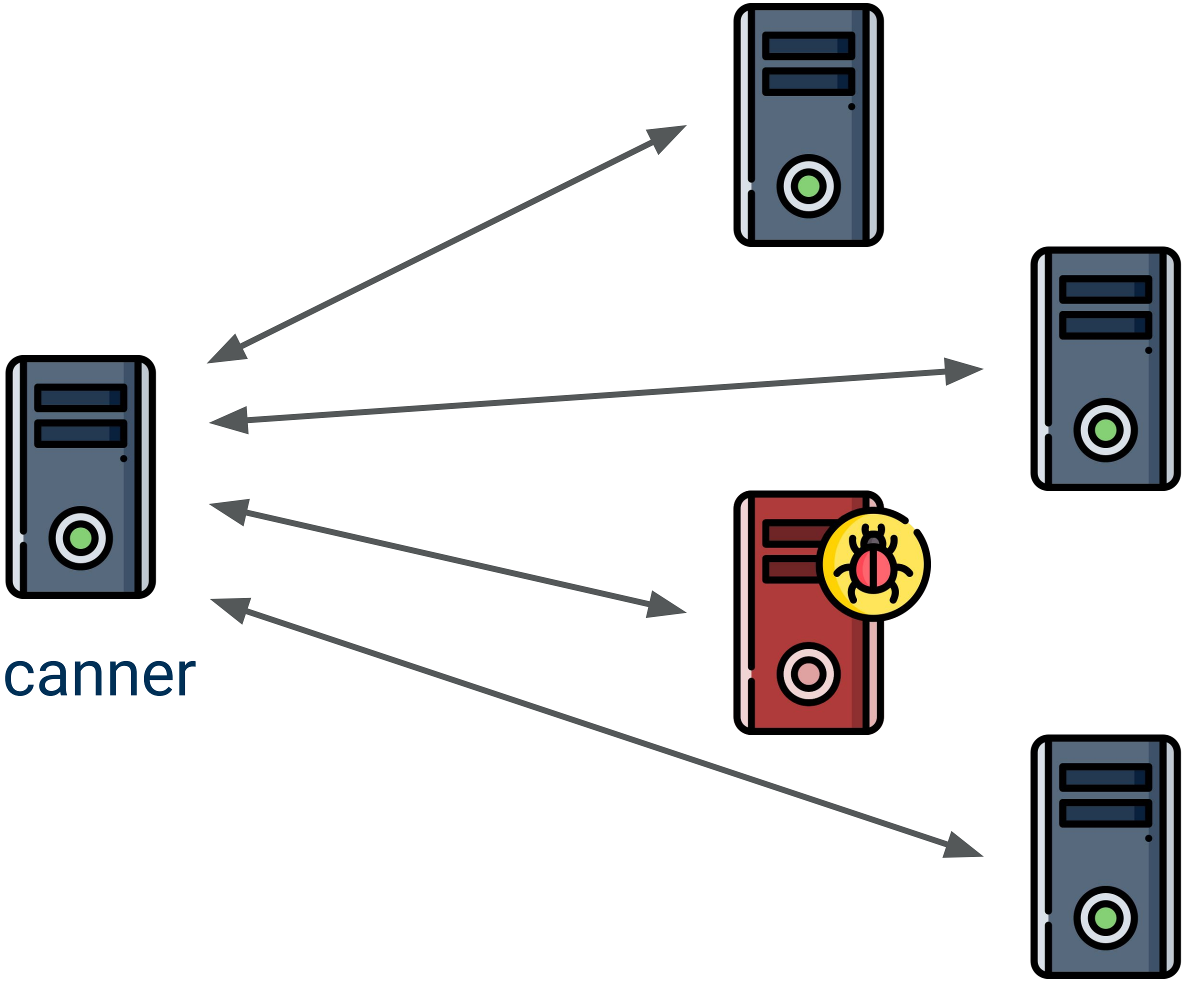
- e
- fort, slow
- ble
- oncerns

Internet Scanning for Malware

For **server-like malware**

1. Try to establish TCP connection with *e.g.*, every IPv4 address

IPv6 scanning is an active research area orthogonal to this work



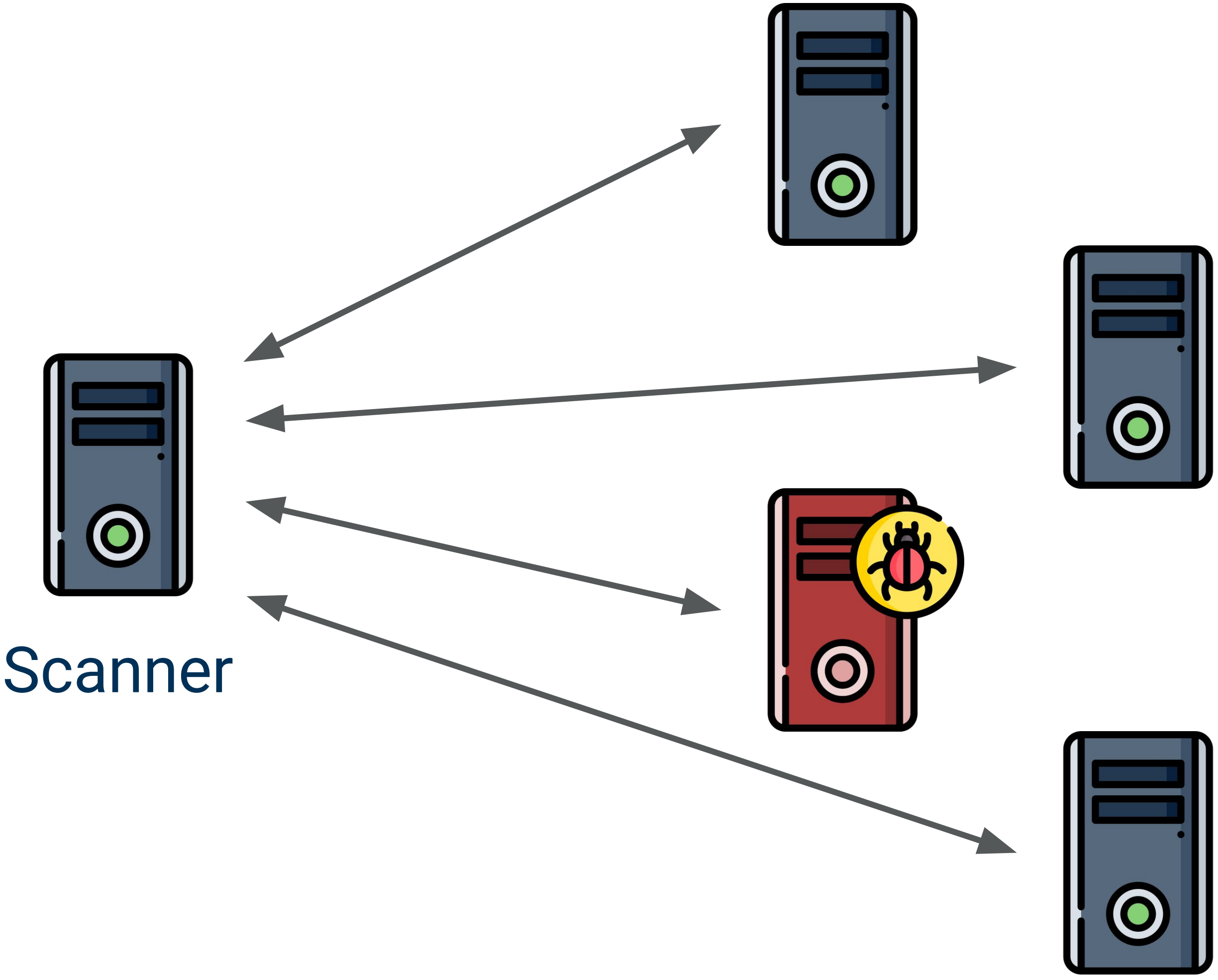
Internet Scanning for Malware

For **server-like malware**

1. Try to establish TCP connection with *e.g.*, every IPv4 address

2. Send **challenge**, receive **response** (**“signature”**)

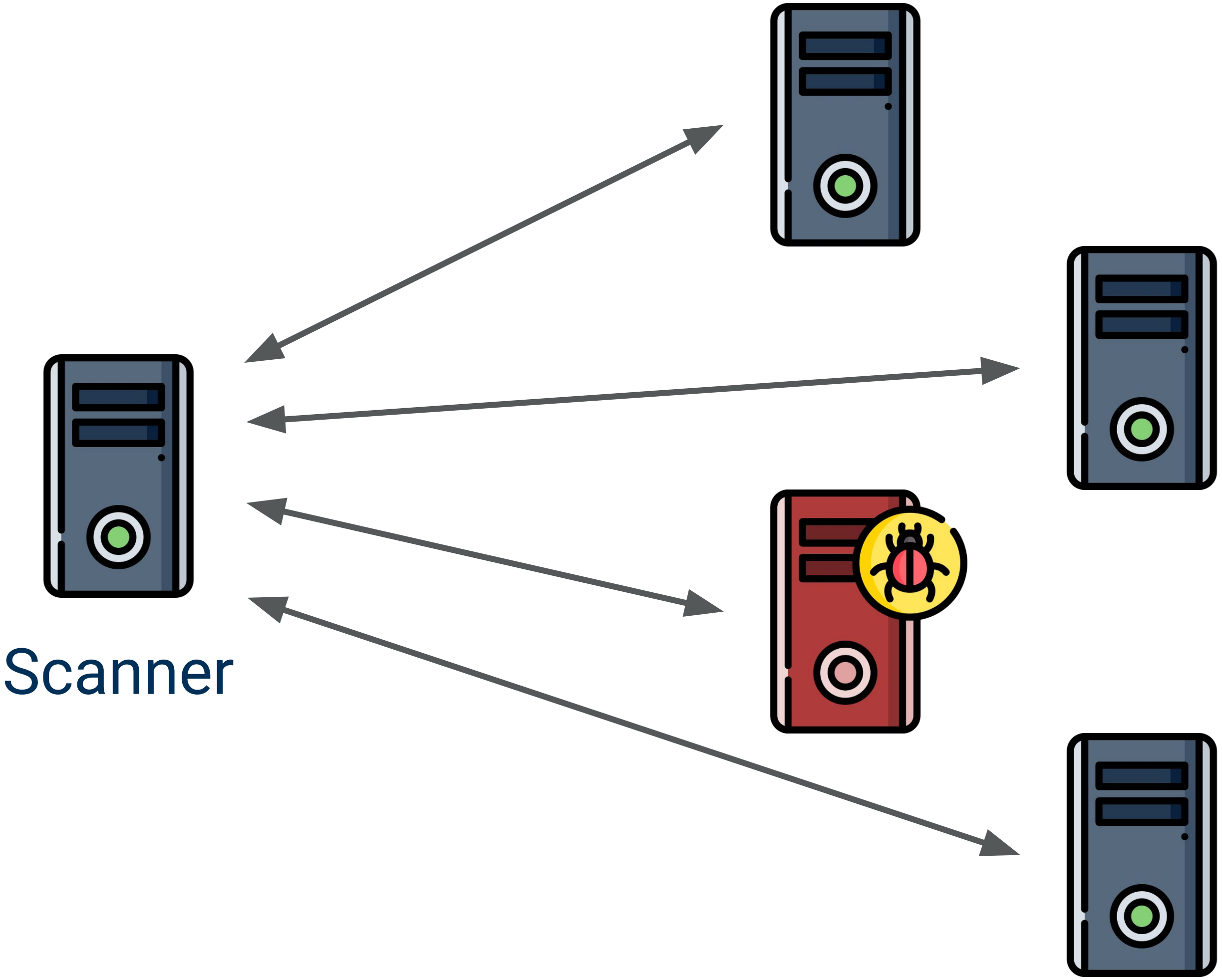
Not to be confused with passive pattern-matching on network traffic



Internet Scanning for Malware

For **server-like malware**

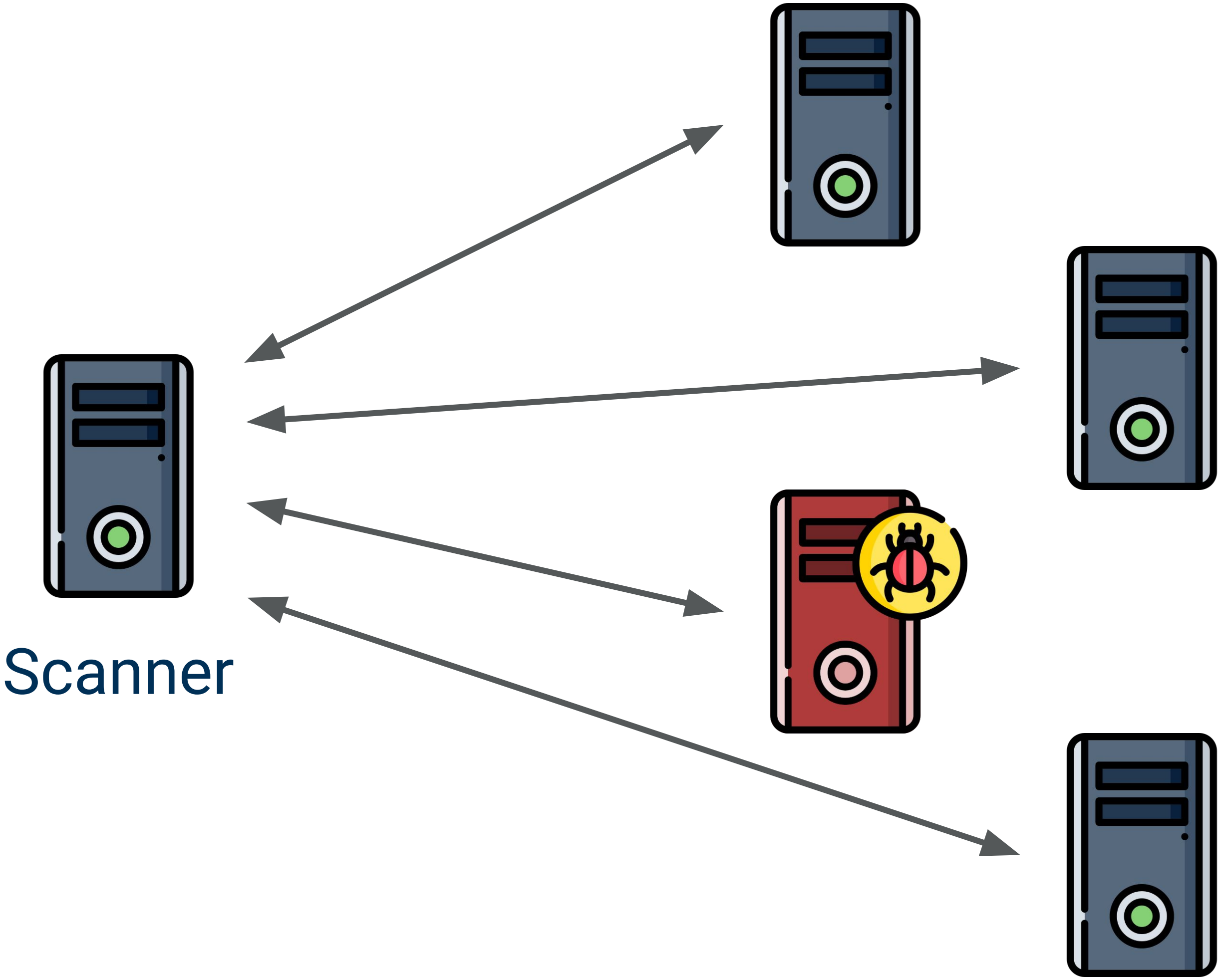
1. Try to establish TCP connection with *e.g.*, every IPv4 address
2. Send **challenge**, receive **response** (“**signature**”)
3. Check whether response appears to be from the malware



Internet Scanning for Malware

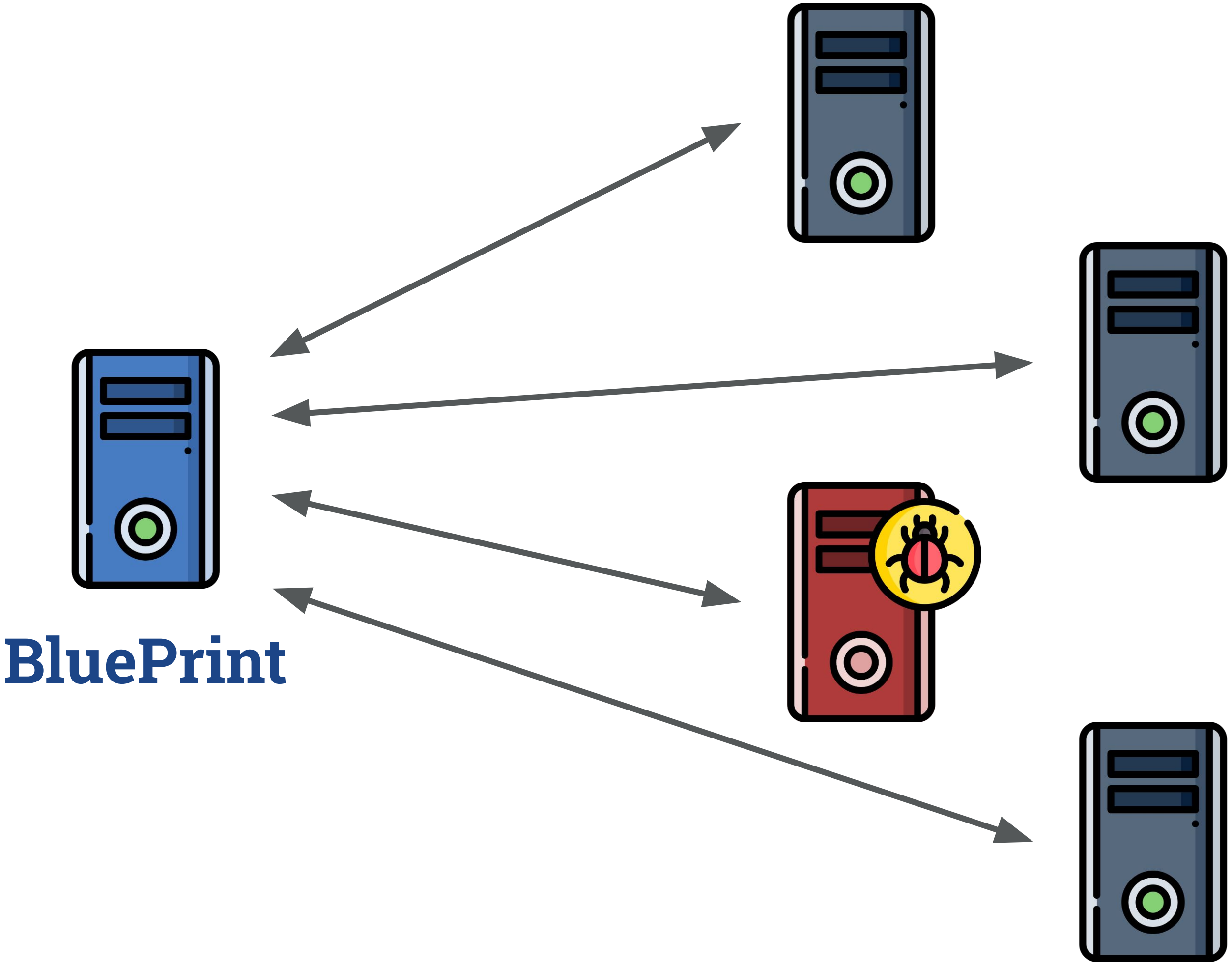
High-performance research-oriented Internet scanning tools (ZMap, ZGrab) are widely used…

…but modules for them are handwritten, one-off, ad-hoc.



Automation: BluePrint

BluePrint is the **first** system able to largely automate the malware scanning process **end-to-end**, from binary analysis to analyzing scan results.



Georgia Tech

Design

Overall Approach

Concrete (Sandboxed) Execution

Overall Approach

☒ No symbolic packet formats

Overall Approach

No symbolic packet formats

Concolic Execution

Overall Approach

❌ No symbolic packet formats

❌ Low path coverage

Overall Approach

✗ No symbolic packet formats

✗ Low path coverage

✓ **Symbolic Execution**

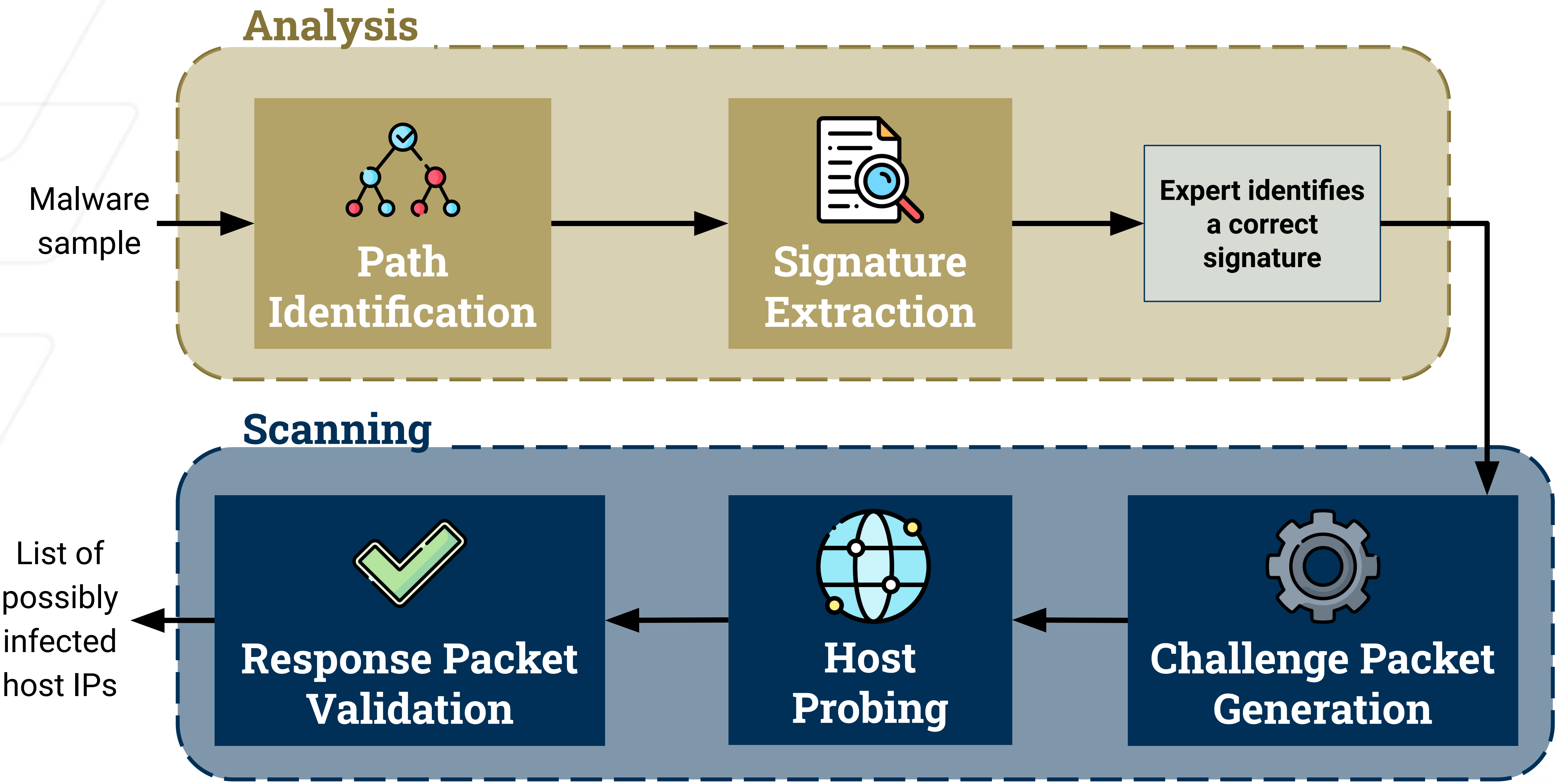
Key Limitations

Malware must listen for connections

Malware must use POSIX socket APIs (*e.g.*, no kernel malware)

Limitations of static and symbolic analysis:

- Obfuscation
- Packed binaries
- Indirect calls



Analysis

Malware sample

Path Identification

Signature Extraction

Expert identifies a correct signature

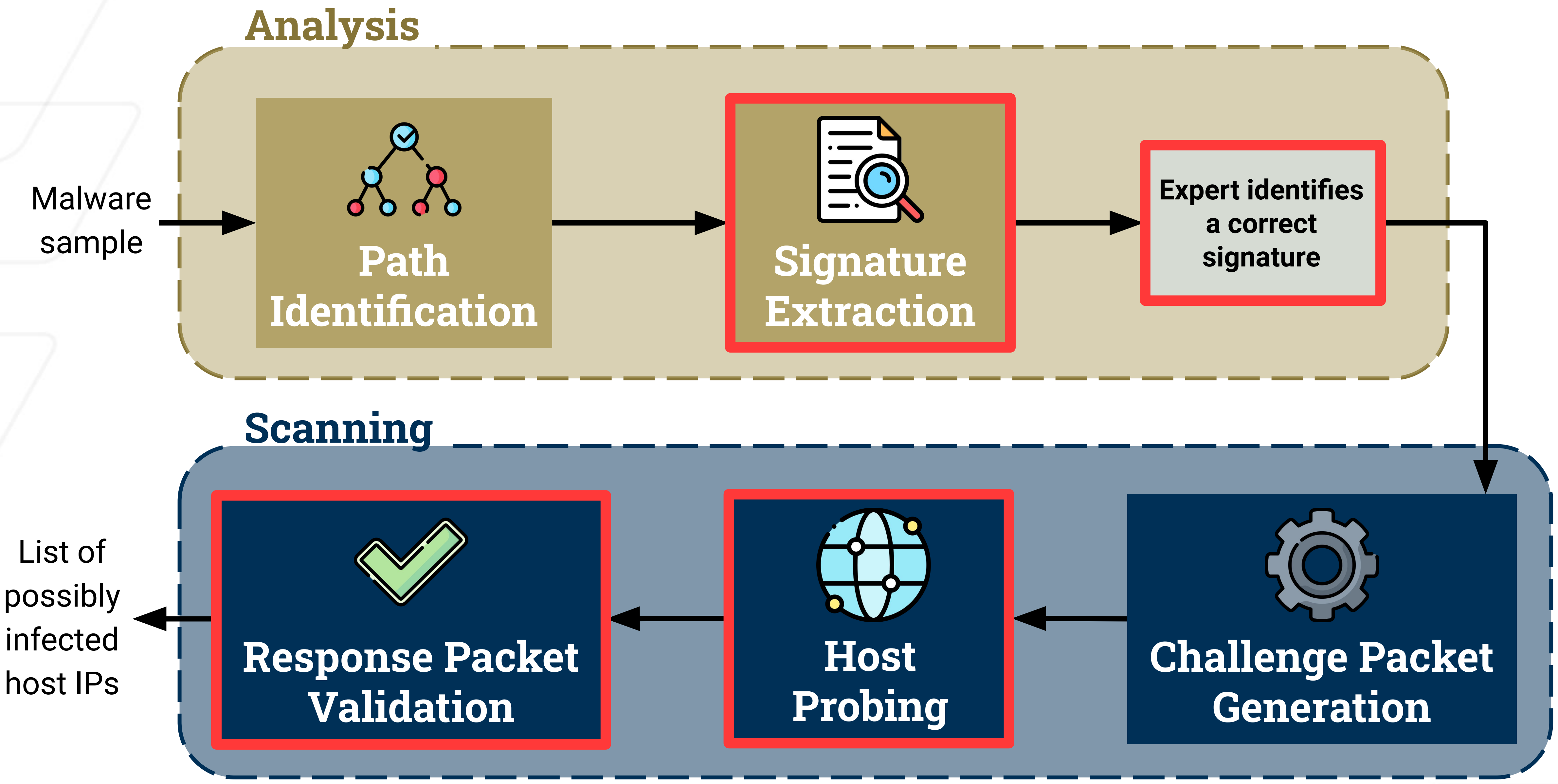
Scanning

Challenge Packet Generation

Host Probing

Response Packet Validation

List of possibly infected host IPs



Analysis

Malware sample

Path Identification

Signature Extraction

Expert identifies a correct signature

Scanning

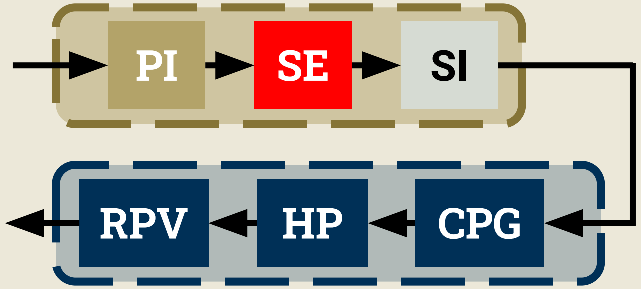
Challenge Packet Generation

Host Probing

Response Packet Validation

List of possibly infected host IPs

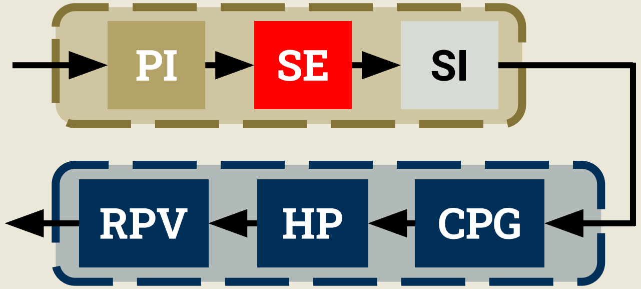
Signature Extraction



Symbolic execution, guided by “path sketches.”

Loose guidance found through static analysis

Signature Extraction

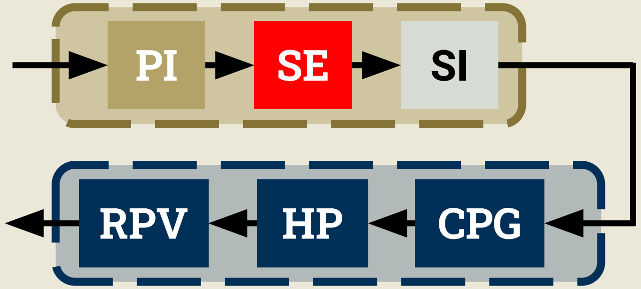


Symbolic execution, guided by “path sketches.”

Goal: Collect constraints on `recv()` and `send()` buffers.

Major Techniques:

Signature Extraction



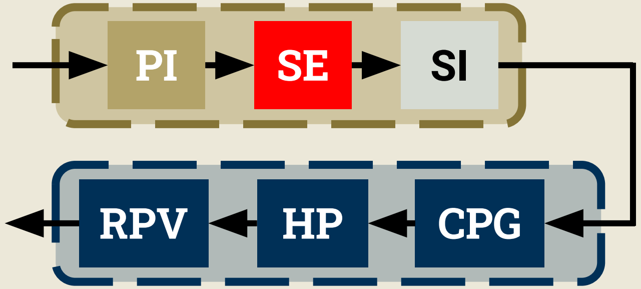
Symbolic execution, guided by “path sketches.”

Goal: Collect constraints on `recv()` and `send()` buffers.

Major Techniques:

Hybrid Exploration

Signature Extraction



Symbolic execution, guided by "path sketches."

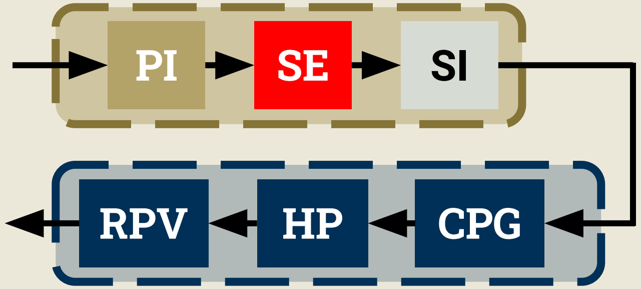
Goal: Collect constraints on `recv()` and `send()` buffers.

Major Techniques:

Hybrid Exploration

Combination of breadth-first search & depth-first search

Signature Extraction



Symbolic execution, guided by “path sketches.”

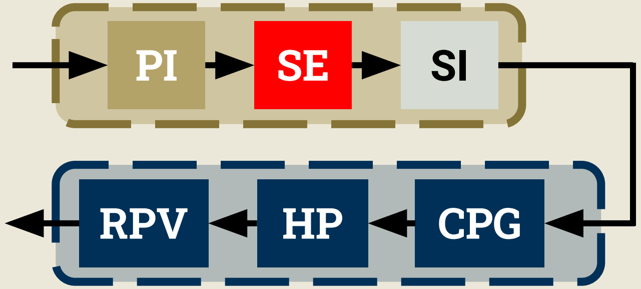
Goal: Collect constraints on `recv()` and `send()` buffers.

Major Techniques:

Hybrid Exploration

Novel Symbolic Models

Signature Extraction



Symbolic execution, guided by “path sketches.”

Goal: Collect constraints on `recv()` and `send()` buffers.

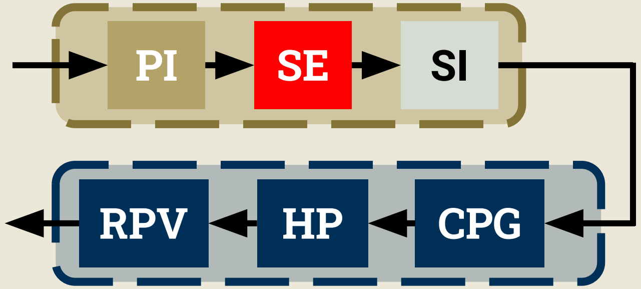
Major Techniques:

Hybrid Exploration

Novel Symbolic Models

Designed to **prevent state explosion** with common networking code patterns

Signature Extraction



Symbolic execution, guided by “path sketches.”

Goal: Collect constraints on `recv()` and `send()` buffers.

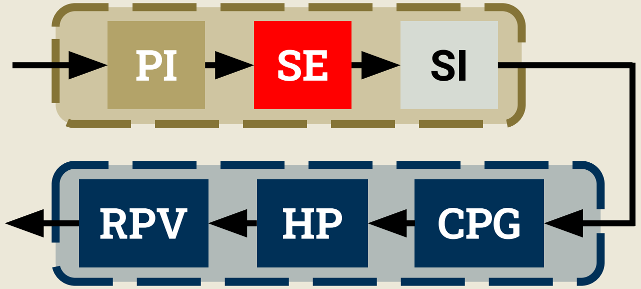
Major Techniques:

Hybrid Exploration

Novel Symbolic Models

Constraint Minimization

Signature Extraction



Symbolic execution, guided by “path sketches.”

Goal: Collect constraints on `recv()` and `send()` buffers.

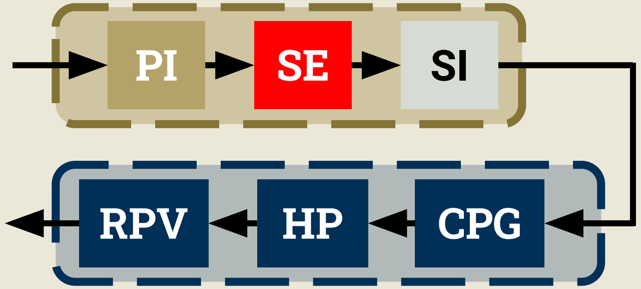
Major Techniques:

Improves performance by **removing irrelevant constraints**

Symbolic Models

Constraint Minimization

Signature Extraction



PI → SE → SI → CPG → HP → RPV

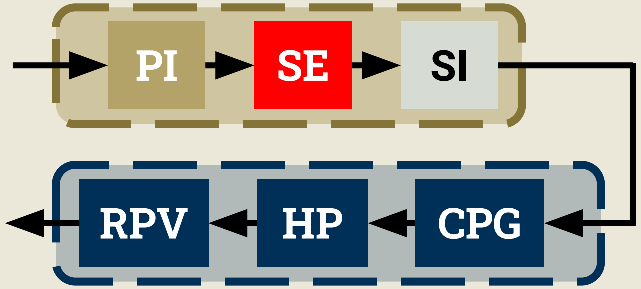
Symbolic execution, guided by “path sketches.”

Goal: Collect constraints on `recv()` and `send()` buffers.

Major Techniques:

Hybrid Exploration	Novel Symbolic Models
Constraint Minimization	Signature Deduplication

Signature Extraction



Symbolic execution, guided by “path sketches.”

Goal: Collect constraints on `recv()` and `send()` buffers.

Major Techniques:

Hybrid Explo

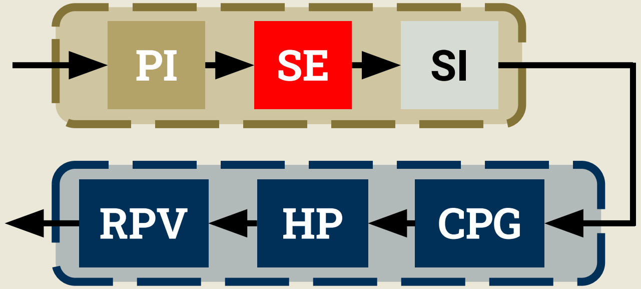
Eliminates duplicate signatures using a **content-aware hashing algorithm**

Constraint Minimization

Signature Deduplication

Georgia Tech

Signature Extraction



Symbolic execution, guided by “path sketches.”

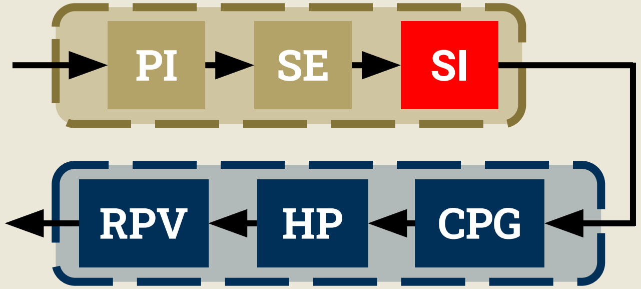
Goal: Collect constraints on `recv()` and `send()` buffers.

Major Techniques:

Hybrid Exploration	Novel Symbolic Models
Constraint Minimization	Signature Deduplication

For *BankShot*: **9 minutes** per sketch, **36** signatures → **12** deduplicated

Signature Identification


PI
SE
SI
RPV
HP
CPG

Expert human analyst selects the best signature. Criteria:

Correct

Signature is not affected by inaccuracies (*e.g.*, concretization)

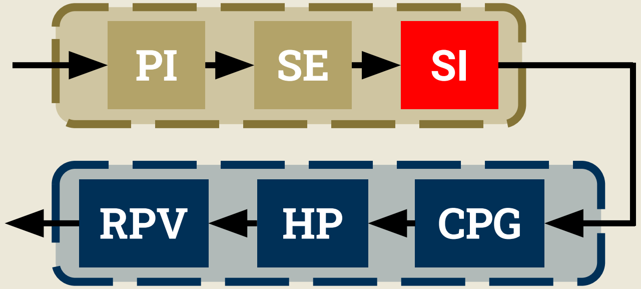
Safe

Signature would not trigger malicious behavior

Distinctive

Signature is different from common protocols

Signature Identification



Expert human analyst selects the best signature. Criteria:

Correct

Signature is not affected by inaccuracies (*e.g.*, concretization)

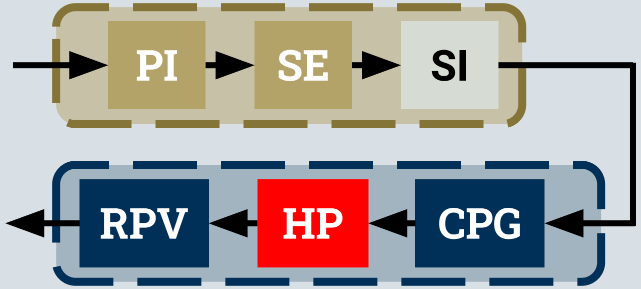
Safe

Signature would not trigger malicious behavior

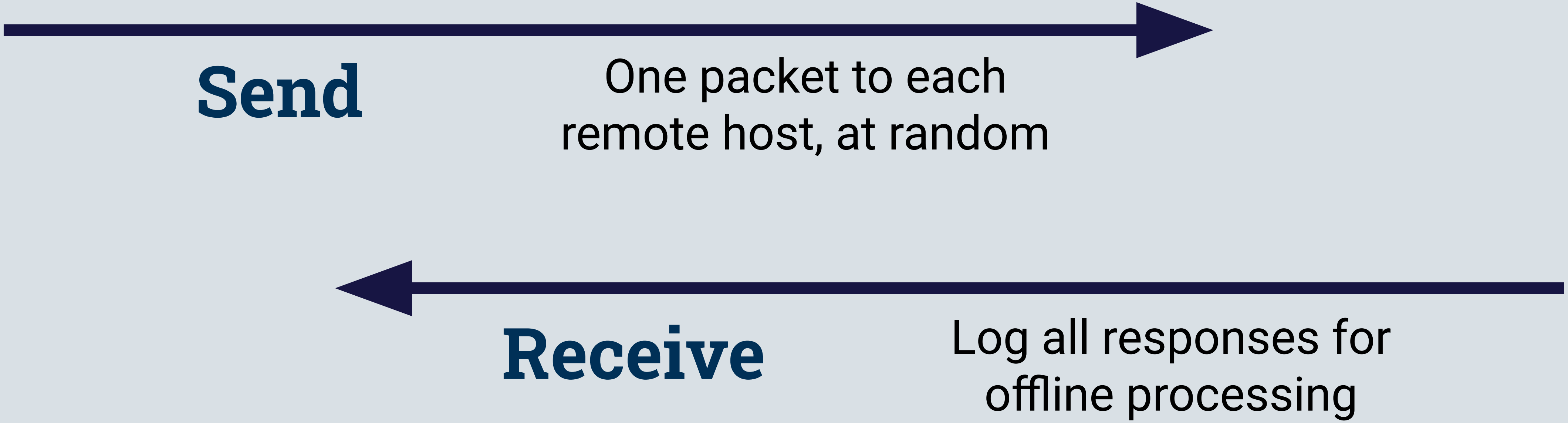
Distinctive

Signature is different from common protocols

Host Probing



High-performance Internet scanning using proven tools (ZMap, ZGrab).



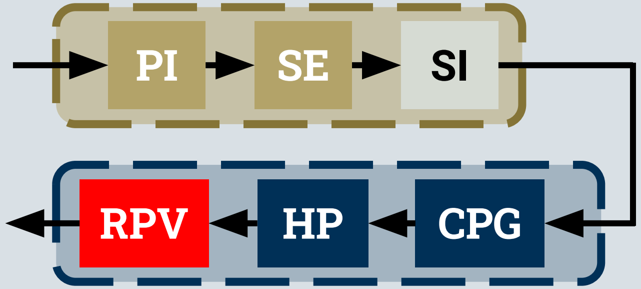
Send

One packet to each remote host, at random

Receive

Log all responses for offline processing

Response Packet Validation



Check constraint satisfiability for each interaction.

Cross-Packet Constraints

Validation may depend on *both* packets

Evaluation

Dataset

Sample	Type	Conn. Listening Purpose	Signature	
			Challenge	Response
BadCall	Proxy	Proxying	Fake TLS ClientHello	Fake TLS ServerHello, Certificate, and ServerHelloDone
BankShot	Proxy	Proxying	XOR cipher seed and ciphered six-byte message	XOR cipher seed and ciphered six-byte message
Derusbi	RAT	C&C	Packet with three ints with “magic” relationship	Packet with three ints with “magic” relationship
FASTCash	RAT	C&C	Fake TLS packet with two ints with “magic” relationship	Fake TLS packet with two ints with “magic” relationship
Gh0st	RAT	Proxying	SOCKS5 handshake: first byte `05`, third `00` or `02`	SOCKS5 handshake: `05 00` or `05 02`
Slingshot	Loader	Payload retrieval	None	`B2 7F 23 43`
Soul	RAT	C&C	None	Fixed HTTP GET header with compressed payload

Signature Accuracy

Generally, BluePrint extracts each signature **accurately**.

Observed inaccuracies stem from:

“Abortive shutdown” (*BadCall*)

Signature mismatch between different malware components (*Derusbi*)

Limitations of symbolic execution (*Soul*)

Efficiency

Sample	Path Ident.	Signature Extraction				Packet Generation
		Time per Sketch		Deduplication		
	Count	Average	Max	Count	Time	Time
Best	1	0:03	0:03	5429 → 16 (0.3%)	< 0:01	0:01
Mean	67	15:26	26:25	38%	0:34	6:36
Median	65	15:39	22:49	17%	< 0.01	0:21
Worst	159	35:59	1:05:57	2 → 2 (100%) 368 → 23 (6%)	3:05	27:05

Effectiveness (Ablation Study)

Sample	P	H	S	F
BadCall	🟥	🟥	🟥	🟥
BankShot	🟥			🟥
Derusbi	🟥			
FASTCash	🟥		🟥	🟧
Gh0st	🟥		🟧	
Slingshot	🟥			
Soul	🟥			🟥

P: Path-sketch guidance

H: Hybrid exploration (BFS + DFS)

S: Symbolic models for `recv()` and `accept()`

F: Static and inline function modeling

red = no signatures produced when disabled

orange = signature quality reduced when disabled

Real-World Scan Results

Discovered **14 real-world *Derusbi* infections.** Reported to law enforcement.

Locations

- India
- Italy
- South Korea
- Sweden
- Taiwan
- USA
- Vietnam

Device Purposes

- Science institute summer internship program website
- University language program website
- Web Feature Service server
- AS CDN

Other samples not found likely due to short or highly targeted campaigns, or running on unusual ports.

Conclusion

Conclusion

BluePrint is the first system to largely automate the end-to-end Internet scanning process for server-like malware, using:

Static Analysis (Path Sketches)

Symbolic Execution

Novel Symbolic Models for Key Network APIs

Proven Internet Scanning Tools

Evaluation demonstrates that BluePrint can successfully analyze and scan for a wide variety of server-like malware.

Thank You

kevin.stevens@gatech.edu

BluePrint is the first system to largely automate the end-to-end Internet scanning process for server-like malware, using:

Static Analysis (Path Sketches)

Symbolic Execution

Novel Symbolic Models for Key Network APIs

Proven Internet Scanning Tools

Evaluation demonstrates that BluePrint can successfully analyze and scan for a wide variety of server-like malware.

Attribution: Icons made by Freepik from www.flaticon.com