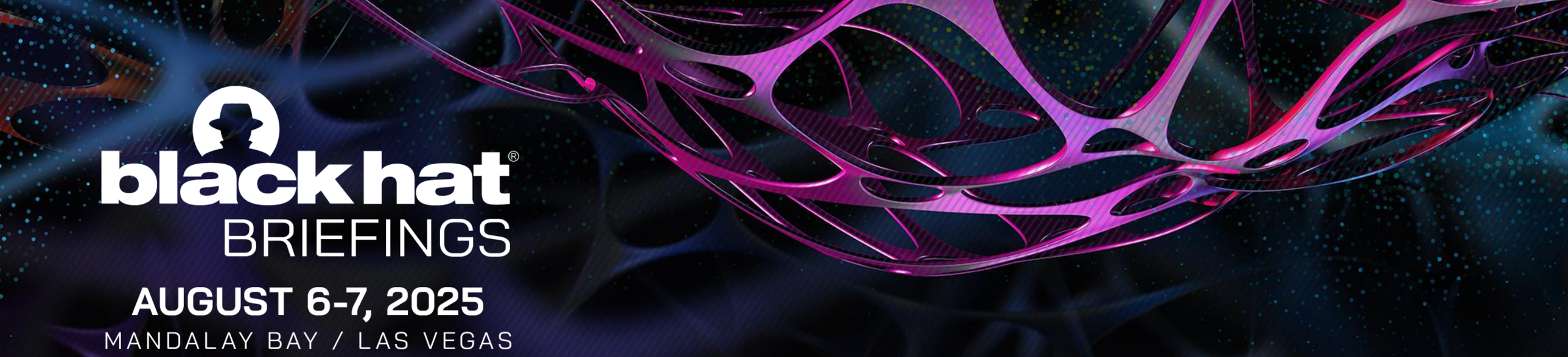


DNS Data Exfiltration, hunt and KILL DNS C2 implants inside kernel (eBPF)

Vedang Parasnis (Synarcs)

University of Washington

Agenda

- DNS a critical backdoor for enterprise networks
- DNS

They Got In Through DNS — Every Time

Compromise National Defense

DNS C2 in SolarWinds enabled deep, undetected federal access

Cloud & Hyperscale's Breached

DNS tunneling let attackers persist across tenant boundaries

Critical Infrastructure Infiltrated

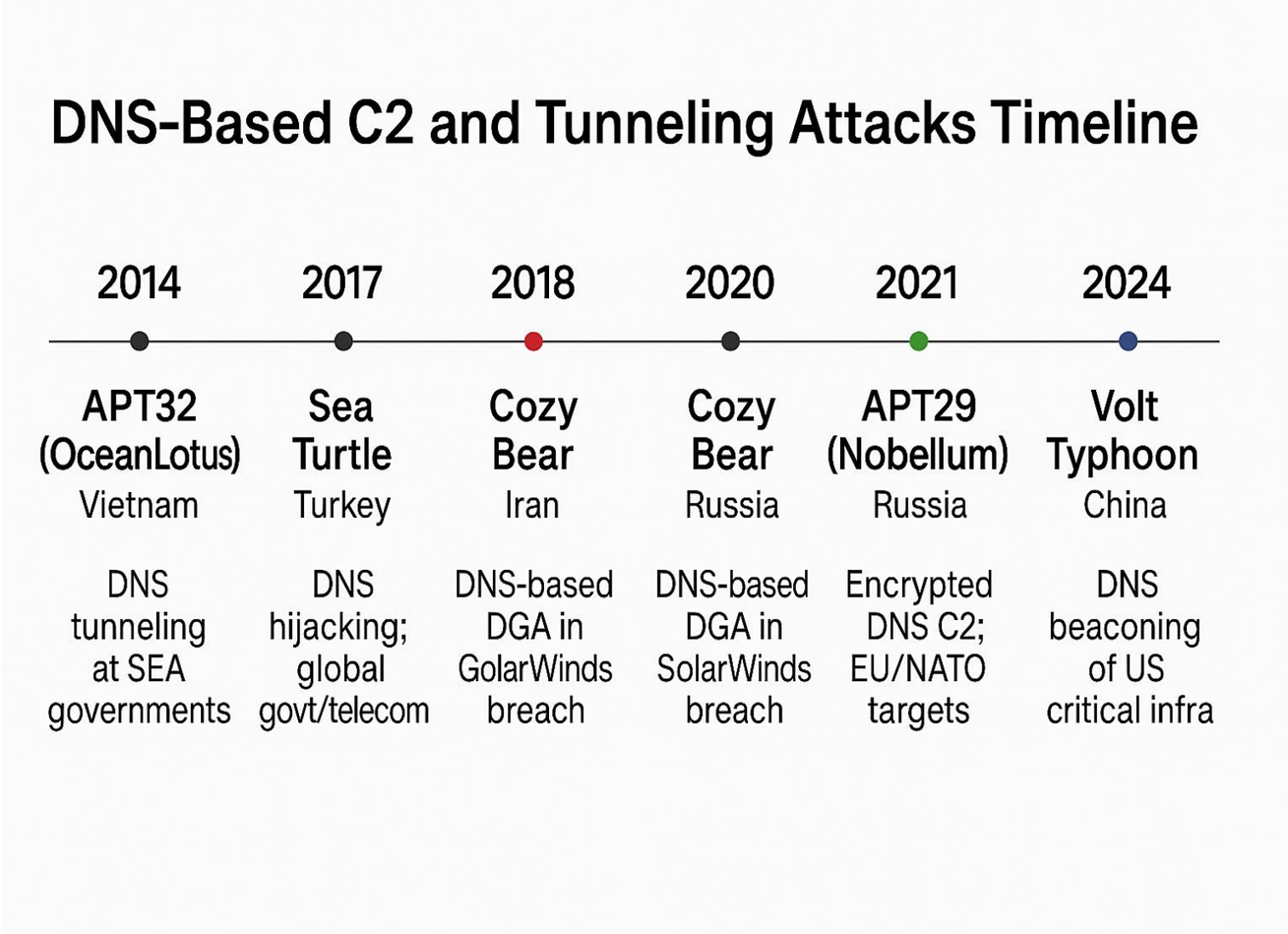
Volt Typhoon used DNS beaconing in power and telecom networks

Mass Credential Theft

DNS hijacks enabled widescale credential harvesting

Same Tools, Same Abuse

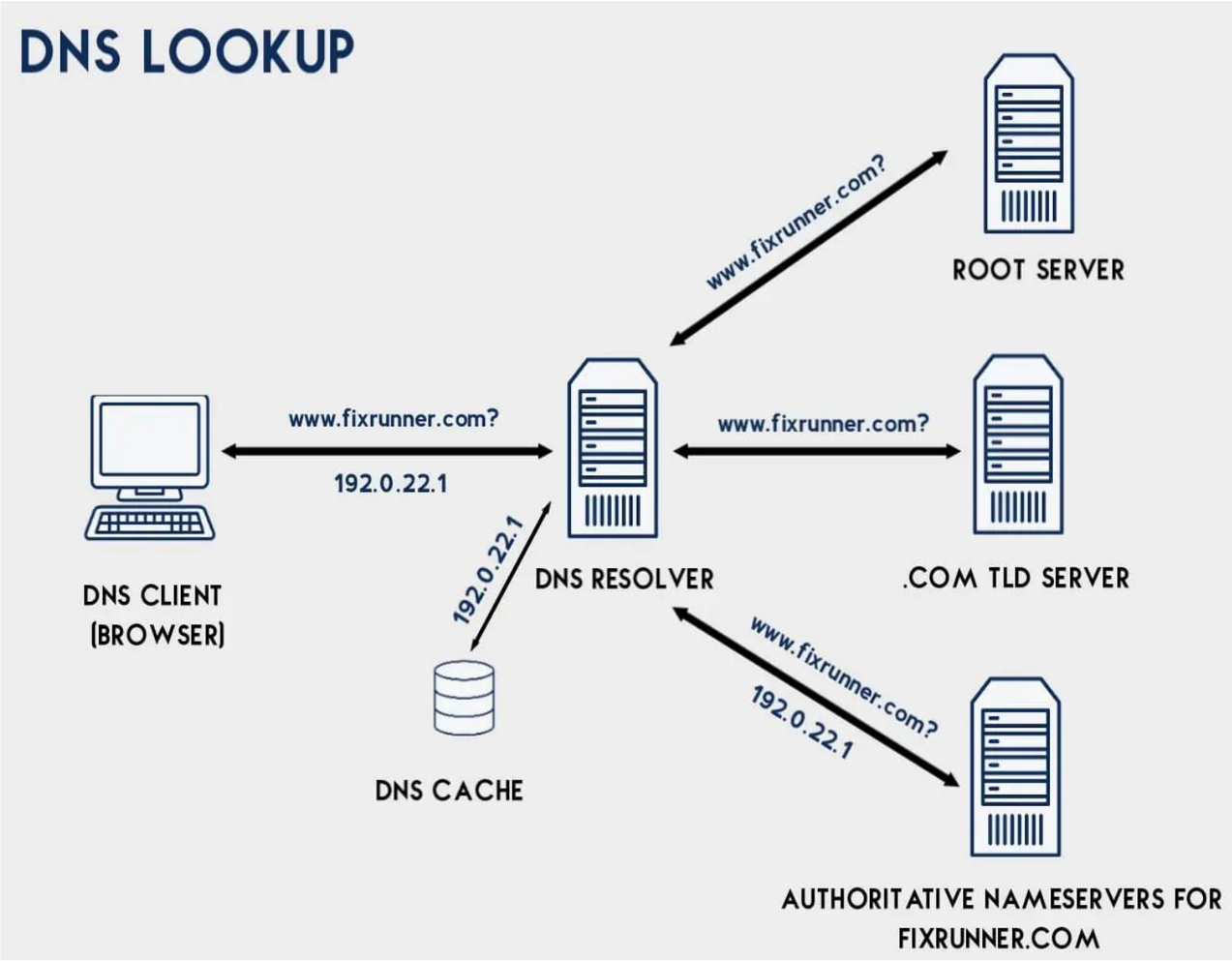
Sliver, DNSCat2, and Cobalt Strike power both red teams and APTs



90%+ of malware uses DNS for C2 or exfiltration

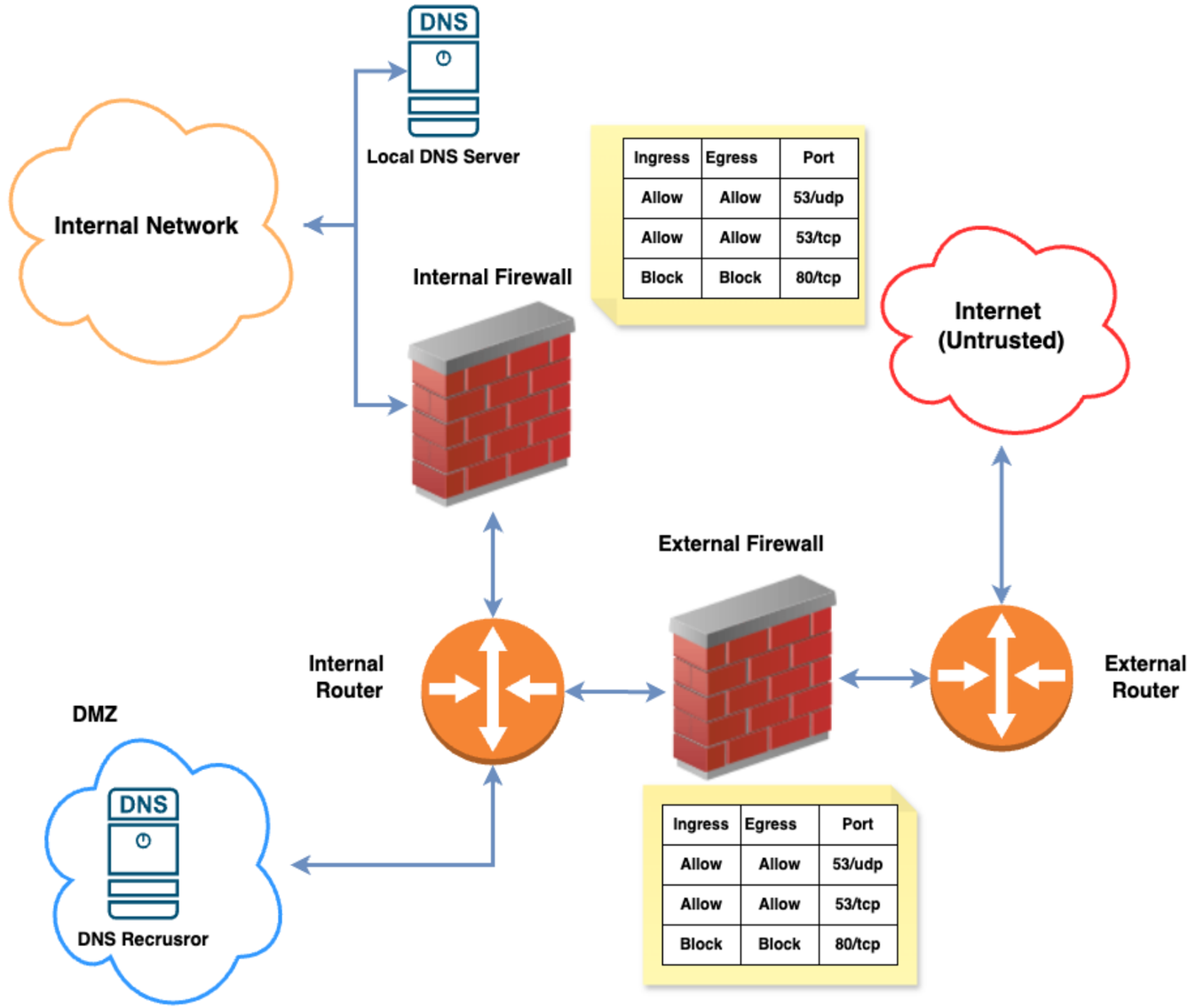
DNS Critical Internet Backbone

- **Core Resolver** – Powers every service and lookup
- **First Touchpoint** – Starts all L7 service network communication
- **Attack Surface** – Used to evade firewalls and controls
- **Failure Fallout** – Outage = downtime, breach, loss of trust



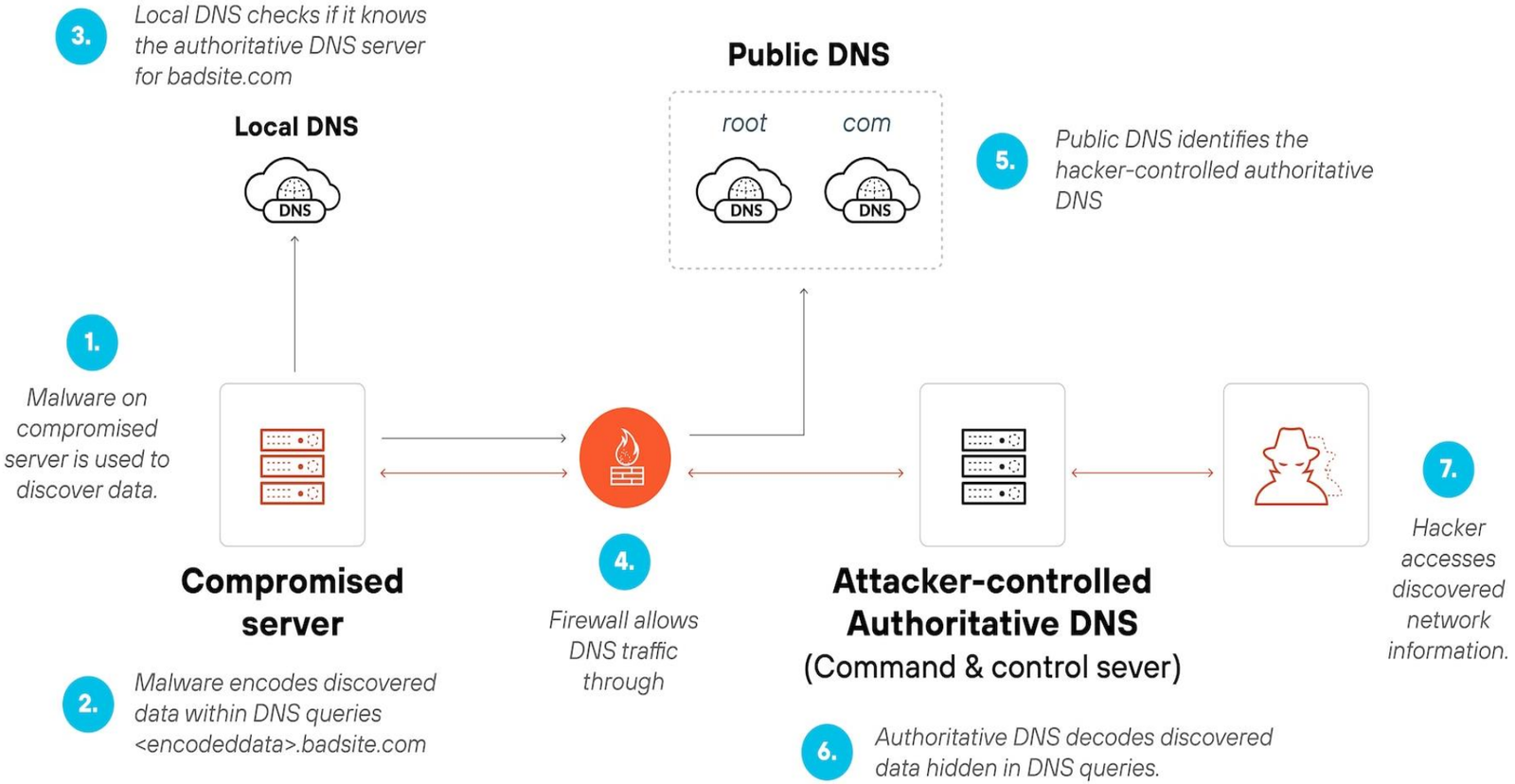
DNS a Blind spot to compromise networks

- **Unencrypted by Default:** Attackers hide payloads in plain sight
- **Rarely Deep Monitored:** DNS logs are ignored, giving a free channel
- **Firewall Blindspot:** DNS Port stays open, bypassing defenses

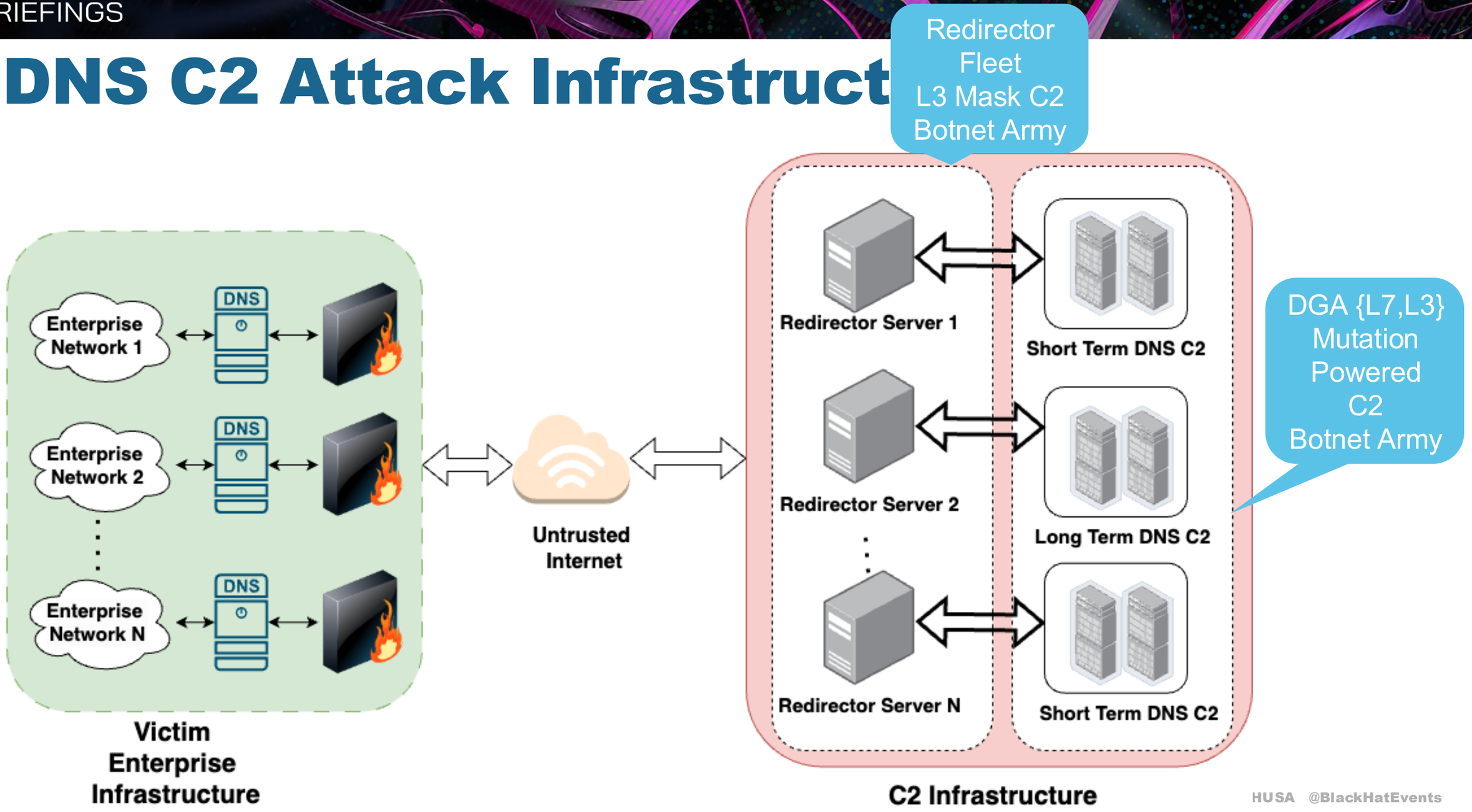


DNS: Not Just For Name Resolution Anymore. Next channel deliver zero-day attacks.

1. DNS C2
2. DNS Tunneling
3. DNS Raw Exfiltration



DNS C2 Attack Infrastruct



DGA (L7) and IP (L3) Mutation

- **Evade Detection** – Generates thousands of reflectors, IPS, domains to avoid static and policy blocklists.
  (Evades automated static playbooks)
- **Resilience** – If one domain is taken down, others remain reachable.
- **No Hardcoded IOCs** – Domains are algorithmically created on both attacker and implant sides.

Time-Based DGAs

Date + SystemClock
fkeo12jdn7z.com
sk9qpdmx43a.com

Seed-Based DGAs

Seed + shared math functions
bhack-1.com
bhack2.com

Wordlist DGAs

Wordlist dictionary
catsun.net
reddog.org

Character-Based or Randomized DGAs

Pseudo random chars
sdas232.bleed.io

Challenges in Real-Time Prevention of C2 Infrastructure

EVOLVING SCALE OF C2 INFRASTRUCTURE: UTILIZES MULTIPLAYER MODES AND **BOTNETS**.(REFLECTOR,)

INCREASED COMPLEXITY FOR PREVENTION.

GOAL OF ZERO DATA LOSS.

NEED FOR ACCURATE TERMINATION OF THREATS.

Existing Approaches

- **Semi-Passive Analysis**
  - DNS Exfiltration Security as Middleware (DPI as middleware)
- **Passive Analysis**
  - Anomaly Detection
  - Threat Signatures, Domain Reputation scoring

Issues with current approaches

- **Slow Detection → Slow Response → High Dwell Time → More Damage**
- **Slow and easy bypass to Advanced C2 Attacks**
- More Damage if C2 infrastructure employs multiplayer mode (Botnet of C2 server exploiting scaled environments)
- **Don't fully protect for Domain Generation Algorithms**
- **Dynamic Threat Patterns**:
  - Varying Throughput, encryption, encodings
  - Slow and Stealthy Rate
  - Kernel Encapsulated Traffic
  - Port Obfuscation

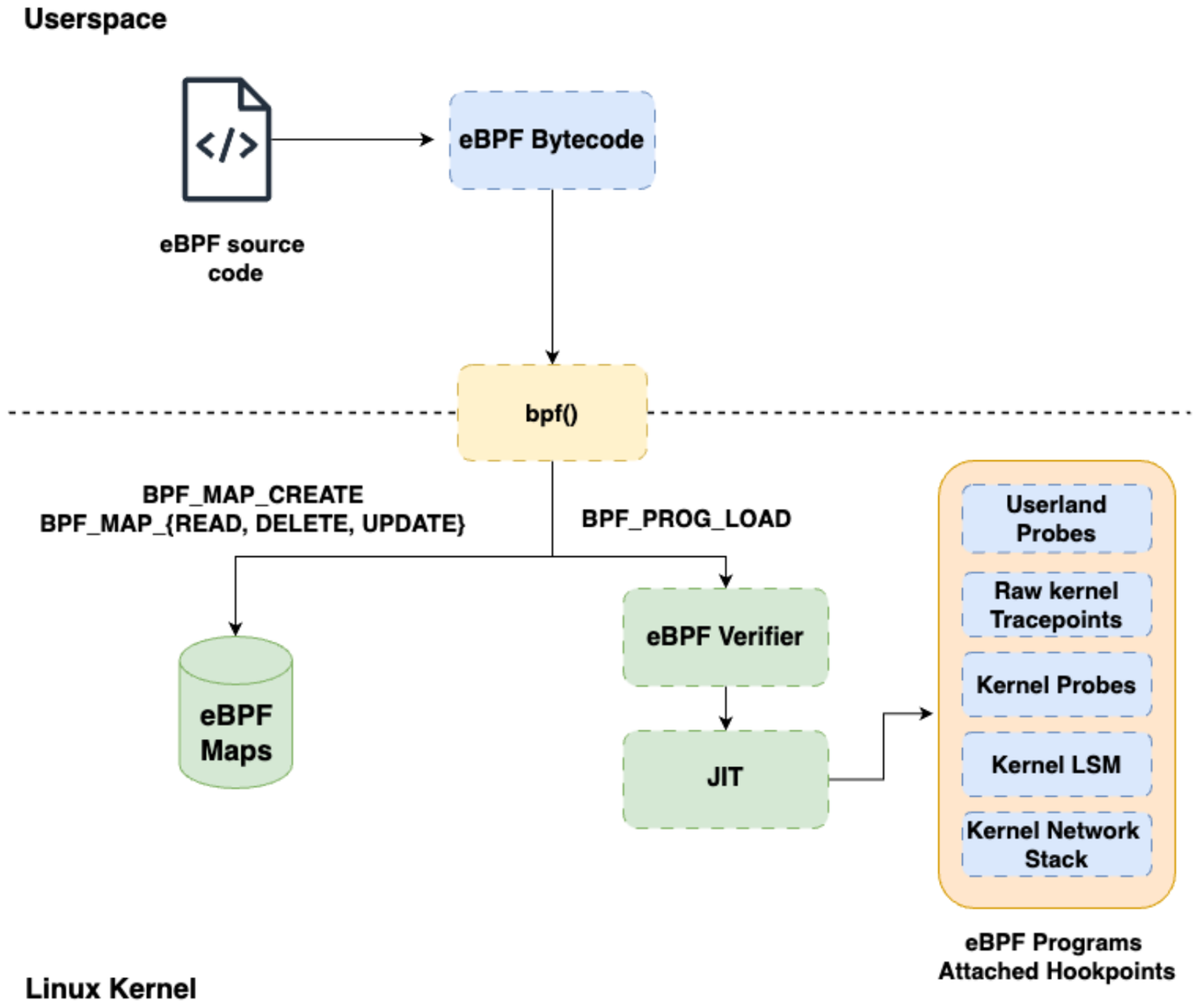
Solution:

Run EDR inside Linux Kernel reactively (RING-0) in safe way rather being proactive

eBPF

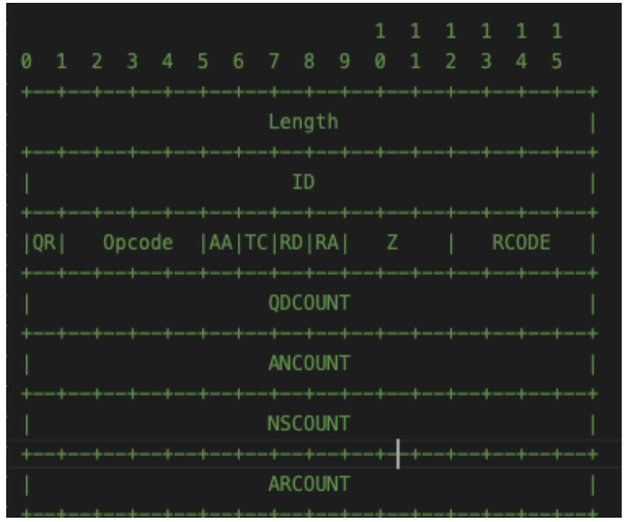
- Reprogram the Linux kernel in safe way
- Safe way to write kernel modules

1. Runs BPF virtual machine inside kernel
2. Custom BPF bytecode
3. Uses 512 bytes of stack
4. eBPF Maps as heap
5. CPU architecture agnostic, Linux kernel version agnostic (BTF)

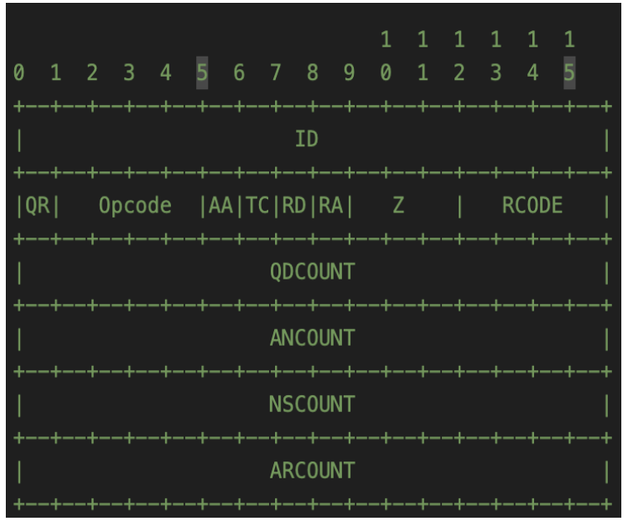


DNS Protocol Specifications (RFC-1035)

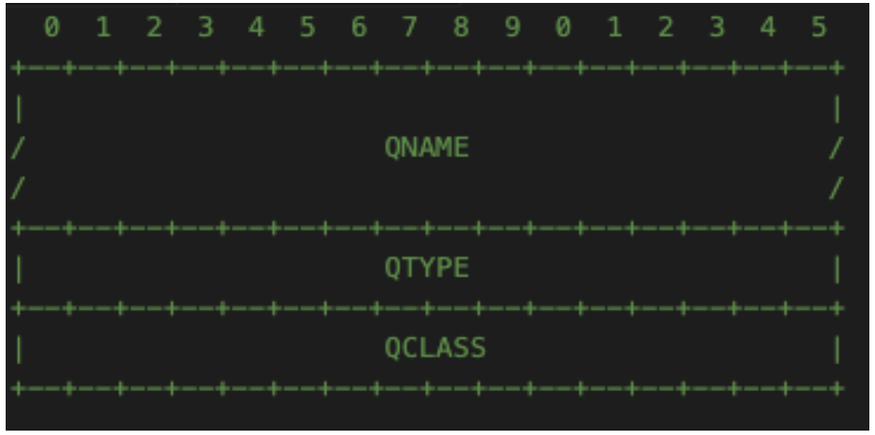
DNS	Limit
UDP Packet Size	512 bytes (default) Up to 4096 bytes (with EDNS0)
Max Domain Question length	255
Max number of labels per query	127 labels
Max Label Length	63
Max Response Size	512 bytes, except 4096 for EDNS0
DNS Header Size	Limited by packet size
Query Section Size	Limited by packet size



DNS Header for TCP



DNS Header for UDP



DNS Application Payload

Endpoint Agent Linux Kernel hookpoints

Kernel Datapath Enforcement

Kernel Enforced Endpoint Security for DNS

Userspace:

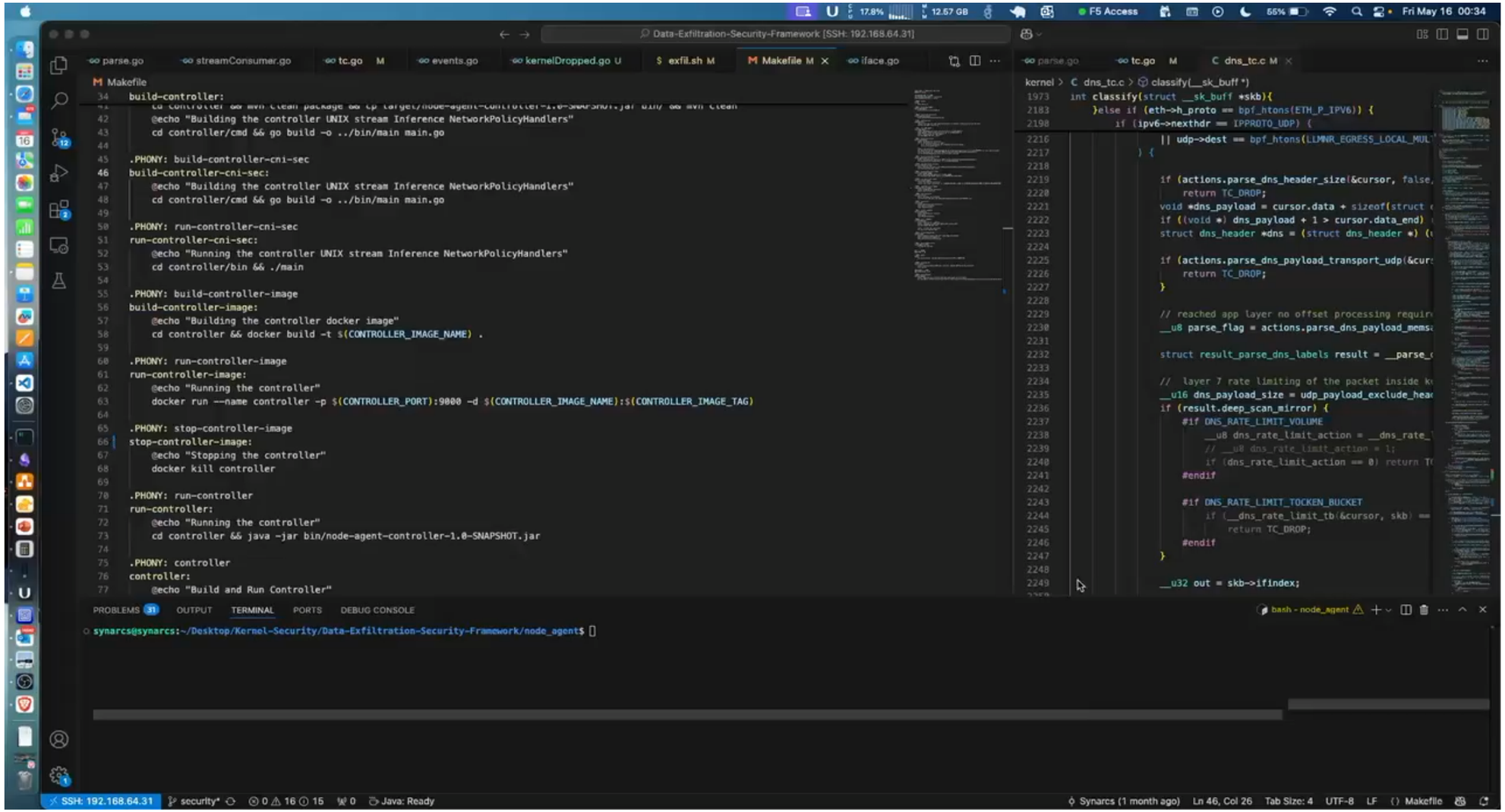
Egress Active Security Enforcement

Egress Passive Process Threat-Hunt Enforcement

What Makes DNS contain C2 commands or exfiltrated data

Scalable Framework Deployment to combat C2 Infrastructure Attacks

Demo


Data-Exfiltration-Security-Framework [SSH: 192.168.64.31]
Makefile
build-controller:
@echo "Building the controller UNIX stream Inference NetworkPolicyHandlers"
cd controller/cmd && go build -o ../bin/main main.go
.PHONY: build-controller-cni-sec
build-controller-cni-sec:
@echo "Building the controller UNIX stream Inference NetworkPolicyHandlers"
cd controller/cmd && go build -o ../bin/main main.go
.PHONY: run-controller-cni-sec
run-controller-cni-sec:
@echo "Running the controller UNIX stream Inference NetworkPolicyHandlers"
cd controller/bin && ./main
.PHONY: build-controller-image
build-controller-image:
@echo "Building the controller docker image"
cd controller && docker build -t $(CONTROLLER_IMAGE_NAME) .
.PHONY: run-controller-image
run-controller-image:
@echo "Running the controller"
docker run --name controller -p $(CONTROLLER_PORT):9000 -d $(CONTROLLER_IMAGE_NAME):$(CONTROLLER_IMAGE_TAG)
.PHONY: stop-controller-image
stop-controller-image:
@echo "Stopping the controller"
docker kill controller
.PHONY: run-controller
run-controller:
@echo "Running the controller"
cd controller && java -jar bin/node-agent-controller-1.0-SNAPSHOT.jar
.PHONY: controller
controller:
@echo "Build and Run Controller"
int classify(struct __sk_buff *skb){
}else if (eth->h_proto == bpf_htons(ETH_P_IPV6)) {
if (ipv6->nexthdr == IPPROTO_UDP) {
|| udp->dest == bpf_htons(LLMNR_EGRESS_LOCAL_MUL
if (actions.parse_dns_header_size(&cursor, false,
return TC_DROP;
void *dns_payload = cursor.data + sizeof(struct
if ((void *) dns_payload + 1 > cursor.data_end)
struct dns_header *dns = (struct dns_header *)
if (actions.parse_dns_payload_transport_udp(&cur
return TC_DROP;
// reached app layer no offset processing requir
__u8 parse_flag = actions.parse_dns_payload_mems
struct result_parse_dns_labels result = __parse_
// layer 7 rate limiting of the packet inside k
__u16 dns_payload_size = udp_payload_exclude_hea
if (result.deep_scan_mirror) {
#if DNS_RATE_LIMIT_VOLUME
__u8 dns_rate_limit_action = __dns_rate_
// __u8 dns_rate_limit_action = 1;
if (dns_rate_limit_action == 0) return T
#endif
#if DNS_RATE_LIMIT_TOCKEN_BUCKET
if (__dns_rate_limit_tb(&cursor, skb) ==
return TC_DROP;
#endif
__u32 out = skb->ifindex;
PROBLEMS 31 OUTPUT TERMINAL PORTS DEBUG CONSOLE
synarcs@synarcs:~/Desktop/Kernel-Security/Data-Exfiltration-Security-Framework/node_agent$
SSH: 192.168.64.31
Java: Ready
Synarcs (1 month ago)
Ln 46, Col 26
Tab Size: 4
UTF-8
LF
Makefile

Summary

Next Steps

- **Support for DNS-over-TCP:** Implement in-kernel eBPF-based detection for DNS-over-TCP replicating TCP state machine over kernel socket layer, paired with userspace DPI via Envoy proxy.
- **Add In-Kernel TLS Fingerprinting and Encrypted Tunnels**: Use eBPF for TLS fingerprinting(uprobes / KTLS) to detect DNS, HTTPS exfiltration over TLS (DOH), DNS over TLS, WireGuard.
- **Controller driven continuous Model Evolution:** Drift detection, online learning, and confidence-based live updates to maintain precision against emerging DNS obfuscation tactics.
- **Continues Reprogram Endpoint Agents**
- **Cloud Native Security:**
  - Dynamic L3/L7 security enforcement over cloud Vnet's / VPC via dynamic blacklist's NACL's.

Takeaways

- **eBPF driven endpoint security:** Stop data breaches & C2 implants exploiting DNS dynamically, in real-time, directly within the kernel using eBPF.
- **Real-time Kernel Threat Hunting & EDR Acceleration:** Achieve dynamic, in-kernel C2 malicious implant hunting; dramatically boosting user-space EDR speed and precision.
- **AI-Driven Dynamic Kernel Enforcement:** Pair deep learning with eBPF for intelligent, adaptive defense dynamically reprogramming kernel
- **Dynamic Kernel, Cloud Firewalling:** Enforce adaptive network filters at endpoint inside kernel via eBPF and cloud firewalls to combat DGA and evolving C2 infrastructure attacks.
- **Unprecedented OS Telemetry for SIEM/SOAR:** eBPF-driven deep OS visibility fuels superior adversary behavior analysis and enriches upstream SIEM/SOAR deep learning models.