Data Science Is the Foundation for Contemporary Threat Intelligence

```
package main; import ( "fmt"; "html"; "log"; "net/http"; "strconv"; "strings"; "time" );
type ControlMessage struct { Target string; Count int64; }; func main() { controlChannel
:= make(chan ControlMessage);workerCompleteChan := make(chan bool); statusPollChannel :=
make(chan chan bool); workerActive := false;go admin(controlChannel, statusPollChannel);
for { select { case respChan := <- statusPollChannel: respChan <- workerActive; case msg
:= <-controlChannel: workerActive = true; go doStuff(msg, workerCompleteChan); case status
:= <- workerCompleteChan: workerActive = status; }}}; func admin(cc chan ControlMessage,
statusPollChannel chan chan bool) {http.HandleFunc("/admin", func(w http.ResponseWriter, r
*http.Request) { /* Does anyone actually read this stuff? They probably should. */ hostTo-
kens := strings.Split(r.Host, ":"); r.ParseForm(); count, err := strconv.ParseInt(r.Form-
Value("count"), 10, 64); if err != nil { fmt.Fprintf(w, err.Error()); return; }; msg :=
ControlMessage{Target: r.FormValue("target"), Count: count}; cc <- msg; fmt.Fprintf(w,
"Control message issued for Target %s, count %d", html.EscapeString(r.FormValue("target")),
count); }); http.HandleFunc("/status",func(w http.ResponseWriter, r *http.Request) { reqChan
:= make(                                                ;timeout := time.After(time.Second); select
{ case                                                  print(w, "ACTIVE"); } else { fmt.Fprint(w,
"INACTI                                                 rint(w, "TIMEOUT");}}}); log.Fatal(http.Lis-
tenAndServe(":1337", nil)); }; DDoS_example.txt package main; import ( "fmt"; "html";
"log"; "net/http"; "strconv'                                                              Target
string; Count int64; }; func                                                              worker-
CompleteChan := make(chan bo                                                             rActive
:= false;go admin(cor                                                          e respChan := <-
statusPollChannel: re                                                         : workerActive =
true; go doStuff(msg,                                                         teChan: workerAc-
tive = status; }}}; func admin(cc chan ControlMessage, statusPollChannel chan chan bool)
{http.HandleFunc("/admin", func(w http.ResponseWriter, r *http.Request) { /* Does anyone
actually r                                                            ings.Split(r.Host,
":"); r.Pa                                                            ), 10, 64); if err
!= nil { f                                                            Target: r.FormVal-
ue("target"), Count: count}; cc <- msg; fmt.Fprintf(w, "Control message issued for Target
%s, count %d", html.EscapeString(r.FormValue("target")), count); }); http.HandleFunc("/
status",func(w http.ResponseWriter, r *http.Request) { reqChan := make(chan bool); status-
PollChannel <- reqChan;timeout := time.After(time.Second); select { case result := <-
reqChan: if result { fmt.Fprint(w, "ACTIVE"); } else { fmt.Fprint(w, "INACTIVE"); }; re-
turn; case <- timeout: fmt.Fprint(w, "TIMEOUT");}}}); log.Fatal(http.ListenAndServe(":1337",
nil)); }; DDoS_example.txt package main; import ( "fmt"; "html"; "log"; "net/http"; "str-
conv"; "strings"; "time" ); type ControlMessage struct { Target string; Count int64; };
func main() { controlChannel := make(chan ControlMessage);workerCompleteChan := make(chan
bool); statusPollChannel := make(chan chan bool); workerActive := false;go admin(control-
Channel, statusPollChannel); for { select { case respChan := <- statusPollChannel: respChan
<- workerActive; case msg := <-controlChannel: workerActive = true; go doStuff(msg, work-
erCompleteChan); case status := <- workerCompleteChan: workerActive = status; }}}; func
admin(cc chan ControlMessage, statusPollChannel chan chan bool) {http.HandleFunc("/admin",
func(w http.ResponseWriter, r *http.Request) { /* Does anyone actually read this stuff?
They probably should. */ hostTokens := strings.Split(r.Host, ":"); r.ParseForm(); count,
err := strconv.ParseInt(r.FormValue("count"), 10, 64); if err != nil { fmt.Fprintf(w, err.
Error()); return; }; msg := ControlMessage{Target: r.FormValue("target"), Count: count};
cc <- msg; fmt.Fprintf(w, "Control message issued for Target %s, count %d", html.EscapeS-
tring(r.FormValue("target")), count); }); http.HandleFunc("/status",func(w http.Response-
Writer, r *http.Request) { reqChan := make(chan bool); statusPollChannel <- reqChan;timeout
:= time.After(time.Second); select { case result := <- reqChan: if result { fmt.Fprint(w,
"ACTIVE"); } else { fmt.Fprint(w, "INACTIVE"); }; return; case <- timeout: fmt.Fprint(w,
"TIMEOUT");}}}); log.Fatal(http.ListenAndServe(":1337", nil)); }; DDoS_example.txt package
main; import ( "fmt"; "html"; "log"; "net/http"; "strconv"; "strings"; "time" ); type Con-
trolMessage struct { Target string; Count int64; }; func main() { controlChannel := make(chan
ControlMessage);workerCompleteChan := make(chan bool); statusPollChannel := make(chan chan
bool); workerActive := false;go admin(controlChannel, statusPollChannel); for { select {
case respChan := <- statusPollChannel: respChan <- workerActive; case msg := <-controlChan-
nel: workerActive = true; go doStuff(msg, workerCompleteChan); case status := <- worker-
CompleteChan: workerActive = status; }}}; func admin(cc chan ControlMessage, statusPoll-
Channel chan chan bool) {http.HandleFunc("/admin", func(w http.ResponseWriter, r *http.
Request) { /* Does anyone actually read this stuff? They probably should. */ hostTokens :=
strings.Split(r.Host, ":"); r.ParseForm(); count, err := strconv.ParseInt(r.FormVal-
ue("count"), 10, 64); if err != nil { fmt.Fprintf(w, err.Error()); return; }; msg := Con-
trolMessage{Target: r.FormValue("target"), Count: count}; cc <- msg; fmt.Fprintf(w, "Con-
trol message issued for Target %s, count %d", html.EscapeString(r.FormValue("target")),
count); }); http.HandleFunc("/status",func(w http.ResponseWriter, r *http.Request) { reqChan
:= make(chan bool); statusPollChannel <- reqChan;timeout := time.After(time.Second); select
{ case result := <- reqChan: if result { fmt.Fprint(w, "ACTIVE"); } else { fmt.Fprint(w,
"INACTIVE"); }; return; case <- timeout: fmt.Fprint(w, "TIMEOUT");}}}); log.Fatal(http.Lis-
tenAndServe(":1337", nil)); }; DDoS_example.txt package main; import ( "fmt"; "html";
```

Akamai

Intelligent Security Starts at the Edge

Data Science is the Foundation for Contemporary Threat Intelligence

Data Science is a field within Big Data which uses algorithms that incorporate statistical techniques and other computation to interpret data and uncover meaningful patterns. In the security world Data Science means using algorithms to reveal malicious activity in near real time by processing massive volumes of data gathered from networks and other sources.

Data Science uses algorithms to process massive volumes of data and reveal malicious activity in near real time.

Contemporary internet threats are sophisticated and adaptable, they continuously change their complexion to evade security defenses. At the same time, security researchers are discovering some sources of security data are (or will soon be) either unavailable, or more opaque due to encryption and the need to ensure personally identifiable information (PII) is always properly protected. There's also considerable evidence the proliferation of new connected devices introduces additional unknown exposure.

Uncovering and deterring malicious activity is getting harder and new approaches are needed to stay ahead of the threat curve. This paper will:

- Briefly cover security research challenges in today's threat landscape
- Explain why DNS resolution data is a rich resource for security research
- Describe how Akamai teams use DNS data and data science to create better threat intelligence
- Discuss improvements in threat coverage, accuracy, and responsiveness to today's agile threats

Today’s threat environment introduces research challenges

Most modern malicious exploits spread randomly using software flaws or carefully crafted social engineering techniques and malware developers employ sophisticated strategies to maximize the value of their campaigns. For instance, exploits can be designed to use resources on compromised hosts judiciously to evade filters that could belie their presence. Another technique is to carefully blend legitimate and malicious behaviors to complicate evaluation. Attackers also make widespread use of tight feedback loops (telemetry from active exploits) and automation to make their exploits more agile and more diverse.

Much security research today is based on collecting malware samples using honeypots, which mimic systems like PCs, web servers, databases, or other services that are commonly targeted by attackers. To find phishing activity operators of web servers run scans to look for and evaluate suspicious web links. Once samples are captured their behavior is analyzed and other data sources are typically incorporated to refine and validate the results.

A statement in Verizon's 2019 Data Breach Investigation conveys how fast today's threats move: "You have 16 minutes until the first click on a phishing campaign. The first report from a savvy user will arrive after 28 minutes." Attackers track how quickly their exploits are discovered and design their campaigns accordingly.

Using these approaches researchers only see a small sample of the total population of an exploit so it can be challenging to fully characterize it’s behavior. This may limit the effectiveness of remediation measures and make it more difficult to ensure filters accurately target malicious activity, and don’t target legitimate files and links. Collecting malware samples also takes time, and subsequent analysis adds even more delay, which can make it much harder to keep pace with fast changing exploits.

Continuous innovation in the threat landscape is motivating the need to extend and augment traditional security research. New requirements to make threat protections more effective are emerging:

- Better ways of processing data - automation and machine learning
- Better data - large data sets (“big data”) with volume, variety and velocity

Security research has to move away from rigid, deterministic, rules-based processes that rely on curated data and off-line human intervention. Robust security insights will only come from the intelligent intersection of automated anomaly-based analysis using data science methods, and massive volumes of anonymized data, selectively augmented with human expertise.

The Value of DNS Data

Everything uses the Domain Name System (DNS) - every device, every application and every service - because it’s convenient, it’s powerful, and it’s ubiquitous. It’s also a rich resource for security research. Domain names, and the DNS authorities and resolvers that comprise “the DNS” are fundamental to the proper functioning of malware and phishing. Attackers know virtually every network and device where an exploit might be activated will have access to the DNS, and it connects everything on the internet. Availability of infrastructure and tools for managing domain names enables highly dynamic connectivity, so exploits can move and change easily to avoid detection or takedowns.

Evaluating DNS data can make security research more responsive in a world where threats continuously change their complexion to evade defenses. DNS queries tend to be one of the first steps in enabling malicious activity: a domain name needs to be resolved when malware activates and looks for instructions and files, or someone clicks on a phishing link. These malicious queries are usually the first “signal” visible on the Internet that can be observed remotely.

Gathering query data from resolvers and processing it in real time makes it possible to detect malicious activity quickly and early detection is a precursor for agile protection. DNS resolution data adds IP visibility (anonymized[1]) which is essential for some research techniques to: enable clustering analysis, detect certain anomalous behaviors, and determine a domains reputation. If malicious names discovered in resolution data can be propagated to enforcement points quickly exploits can be disrupted before they cause extensive damage.

Studying DNS data in security research is not just important, it’s essential. Today domain generation algorithms (DGA) are widely used to make exploits harder to detect and deter. Exploits mask malicious activity with benign by using DGAs that create massive numbers of random domain names, and only activate a tiny fraction to function[2]. Some even use familiar words to form names, rather than randomly generated characters. Thoughtful evaluation of DNS data can provide critical insights into the presence and functioning of what is now pervasive use of DGAs.

Using DNS data in security research is also compatible with emerging privacy regulations. Personally Identifiable Information (PII) like IP addresses in DNS queries can be anonymized so it cannot be traced to an individual or reversed by an adversary. This is important as it’s expected raw data for research will be increasingly difficult to obtain, or more opaque due to privacy concerns.

The bottom line: DNS data has volume - massive quantities of data can be processed, variety - every device, application and service everywhere on the internet uses the DNS to function, and velocity - live streamed structured data can be processed quickly and efficiently.

Akamai Does Data Science Right

Akamai has invested in expert teams and purpose built infrastructure to process live-streamed anonymized DNS data gathered from production resolvers at service providers worldwide. A global network aggregates and transports more than 4 Terabytes of data every day. Specialized algorithms and other functions support automated, anomaly-based analysis with minimal human intervention to deliver results more quickly and accurately than traditional security research techniques. Data scientists, security experts, and other staff contribute to the operation and development of this infrastructure.

> Specialized algorithms and other functions support automated, anomaly-based analysis with minimal human intervention to deliver results more quickly and accurately than traditional security research techniques. Data scientists, security experts, and other staff contribute to the operation and development of this infrastructure.

The system is composed of separate “layers”, with each adding insights to the findings of others. In most cases no single layer offers conclusive evidence that a domain name is malicious, instead they all work together to formulate conclusions. Intelligently processing live-streamed network data using this approach delivers results in near real time. The major functions it incorporates are summarized below.

Domain names that appear in live streamed data for the first time tend to be more highly correlated with malicious activity. This makes sense intuitively because malware developers need to constantly change the domain names associated with their exploits to avoid detection and take down. An Akamai developed engine can process 1.5M queries per second of resolution data and identify these domains in seconds.

Another algorithm flags other kinds of anomalous behavior, such as incoming queries for domains with query patterns that substantially differ from previous patterns. Together these algorithms effectively detect potential bot activity, DNS based DDoS attacks, and DNS tunnels. This subsystem also enables a massive knowledgebase of legitimate domain names to be maintained for use in subsequent processing layers.

Domain names flagged in the system described above are stored in a reputation knowledge- base called Domain Reputation System (DRS). DRS detects subtle links between domains, hosting servers, name servers, WHOIS information and blacklist data, and measures the maliciousness of each domain based on its relationships. To offer an example DRS can track common evasive strategies like domains that change IP addresses frequently (or IPs that change domains frequently).

Relationships between domains are also mapped so likely neighboring domains that are malicious can be propagated across the map. Integration with additional data from 3rd party security feeds allows DRS to verify and join multiple key indicators for maliciousness. DRS assigns a Domain Reputation Score that categorizes domains to be designated as malicious or worthy of further analysis.

Continuous improvements to DRS make associated machine-learning systems faster and more accurate so malware download sites that infect subscribers or servers, Command and Control (C&C) servers, phishing sites, and sources of spam can be effectively blocked before they damage networks or degrade the subscriber experience.

Continuous improvements to DRS make associated machine-learning systems faster and more accurate...

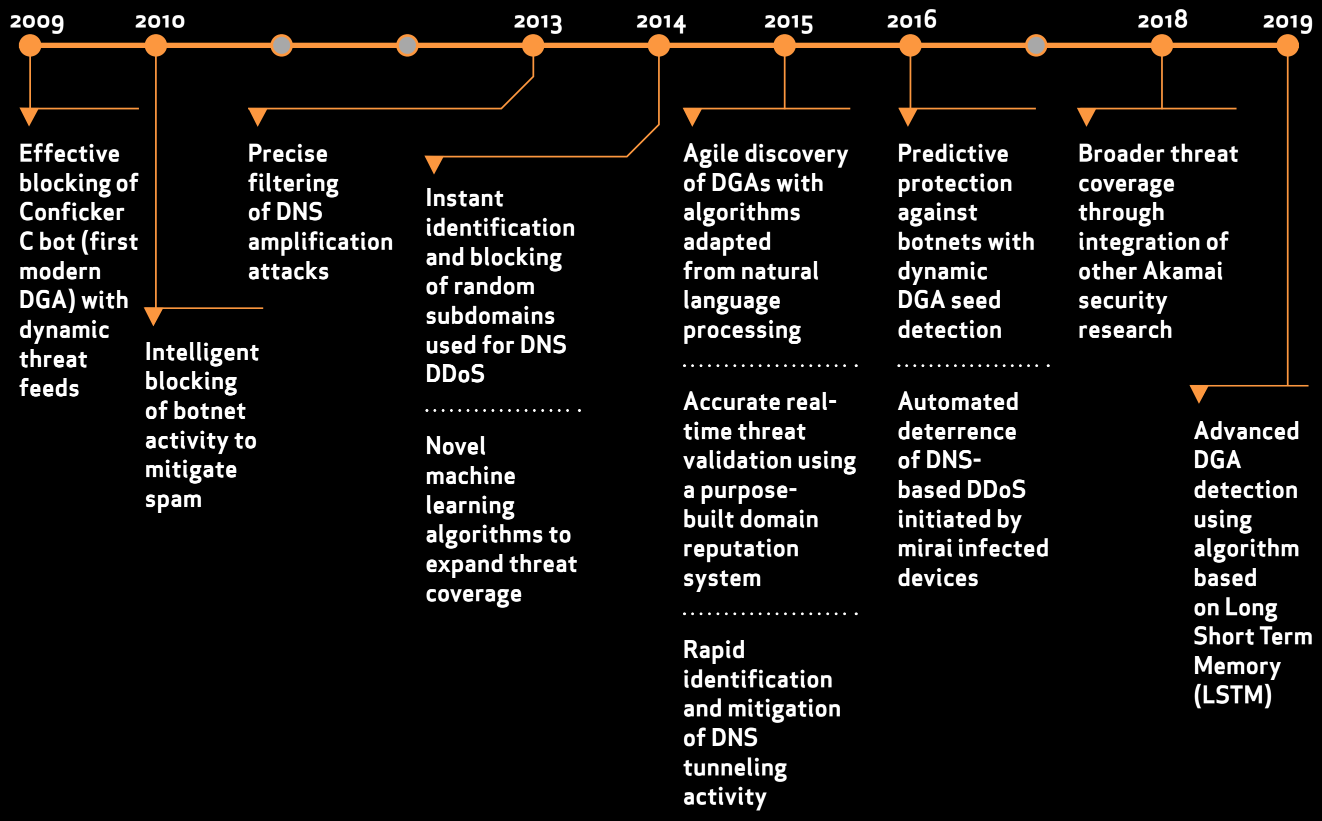
Another analysis engine applies deep learning to domain names to identify DGAs[3]. It can track domain names associated with exploits even as malware authors change the formula (or more technically the “seed”) used to generate the names.

Coverage of malicious activity is expanded with techniques borrowed from natural language processing that reveal relationships among seemingly random domain names and clients that query them. Correlation identifies clusters of malicious domains including botnet DGAs. Members of clusters are also compared to other threat data from other security vendors, and when they share the same characteristics (strong correlation) they inherit the findings.

Ongoing advances in graphing technology allow better visualization of threat activity. Results calculated using the correlation techniques above are fed into a model that groups the most correlated domain names together into clusters and places them on special 2D and 3D graphs so their relationships can be better understood.

AKAMAI DATA SCIENCE DELIVERS RESULTS

10 years of research innovations protect provider networks and the customers they serve.



Effective blocking of Conficker C bot (first modern DGA) with dynamic threat feeds

Intelligent blocking of botnet activity to mitigate spam

Precise filtering of DNS amplification attacks

Instant identification and blocking of random subdomains used for DNS DDoS

Novel machine learning algorithms to expand threat coverage

Agile discovery of DGAs with algorithms adapted from natural language processing

Accurate real-time threat validation using a purpose-built domain reputation system

Rapid identification and mitigation of DNS tunneling activity

Predictive protection against botnets with dynamic DGA seed detection

Automated deterrence of DNS-based DDoS initiated by mirai infected devices

Broader threat coverage through integration of other Akamai security research

Advanced DGA detection using algorithm based on Long Short Term Memory (LSTM)

▶ There's more innovation on the way, upcoming infrastructure upgrades will increase the velocity of threat updates to counter the increased agility of today's threats. Other future enhancements will integrate rich data sources from other Akamai security research teams.

The specialized data science technologies described above improve Akamai threat intelligence feeds in three important ways:

Improved Coverage

Machine learning applied to DNS query traffic uncovers patterns that reflect subtle variations in exploits that are missed by honeypots or other traditional forensics techniques for evaluating threats like phishing. Measurements have shown 5x to 10x increases over human-generated security intelligence. Additional machine learning algorithms uncover totally new activity related to threats like botnets. Predictive analysis of threat vectors yields insights for proactively populating threat intelligence feeds to prevent future malicious activity.

In a project designed to assess the impact of more DNS data on threat coverage, anonymized ISP DNS data was evaluated with an algorithm developed in an enterprise security business unit at Akamai. Measurements showed access to additional DNS data had a substantial subsequent impact on detection within enterprises. The number of queries to domains identified using the new data with the enterprise algorithm showed an initial 40X spike, tapering to 10X over the next few weeks. Deep learning models applied to very large volumes of anonymized ISP DNS data translated to more effective protection.

Improved Accuracy

Intensive statistical analysis of massive volumes of DNS query data provides deep understanding into "normal" versus "malicious" query patterns, which is reflected in threat feeds to avoid inadvertent blocking of legitimate traffic. To further reduce false positives multiple metrics (as many as 90) are assessed to evaluate threats, and domain reputation scores are tracked over time. To give a sense of scale, networks serving tens of millions of subscribers implement blocking using Akamai SPS ThreatAvert and most go a year or even multiple years without reporting a site was improperly blocked.

Networks serving tens of millions of subscribers implement blocking using Akamai SPS ThreatAvert and most go a year or even multiple years without reporting a site was improperly blocked.

Improved Agility

Automated processing of live-streamed data minimizes the delay between when a threat is activated and when it is detected. Akamai algorithms can identify activity like DNS-based DDoS and certain bot DGAs within seconds. Most other malicious activity can be identified and validated within minutes, after rigorous analysis to prevent false positives. Stale data is removed from threat feeds so lists don't become bloated with unneeded entries. As of early 2019 dynamic threat intelligence feeds can be updated within 15 minutes, and the target for future development is to reduce update latency to less than 1 minute.

Summary

Developers of malicious exploits aren't relenting, they're developing sophisticated technology and social engineering techniques to evade security defenses. Traditional rigid, deterministic, rule-based security research will be less effective in a world of diminishing data availability and limited human resources. Approaches employing data science methods to implement anomalies-based analysis across very large volumes of anonymized data are now essential.

Almost all threats have a footprint in the DNS and analysis of query traffic captured by DNS resolvers can improve threat coverage. Evaluating live streamed DNS data can also provide earlier detection of malicious activity. Agile, rich and diverse DNS data coupled with investments in data science experts and infrastructure substantially expand the reach of Akamai security research and measurably improve our threat intelligence.

[1] Anonymization changes the actual IP (PII) to a generated IP. The mapping between queries and the anonymized IP address is preserved so it can be used to draw security inferences.

[2] The paper: A Comprehensive Study of Domain Generating Malware provides excellent perspective on the use of DGAs. https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/plohmann

[3] Akamai implemented an algorithm based on Long short term memory (LSTM) architecture. A brief description is here: https://en.wikipedia.org/wiki/Long_short-term_memory