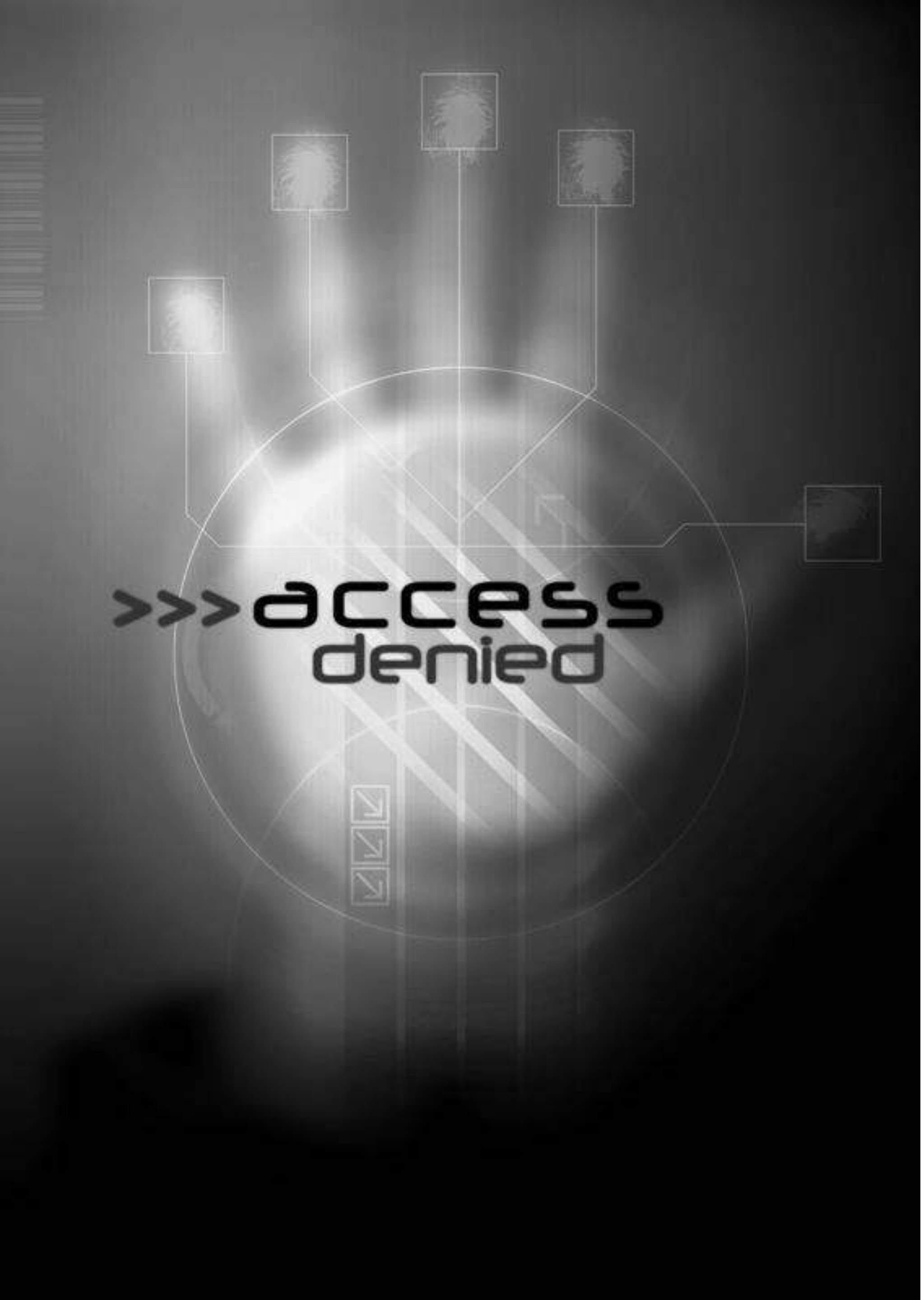


Real-time Computational Intelligence Protection Framework Against Advanced Persistent Threats

Democritus University of Thrace
Dep. of Forestry & Management of the Environment & Natural Resources
Lab of Forest-Environmental Informatics & Computational Intelligence

Konstantinos Demertzis – Lazaros Iliadis

1. **Introduction**
2. **Dataset**
3. **Methodologies**
4. **Results**
5. **Conclusions - Future Directions**
6. **References**
7. **Questions - Discussion**

What is an Advanced Persistent Threat?

✓ *“An Advanced Persistent Threat (APT) is a set of stealthy and continuous computer hacking processes in which an unauthorized person gains access to a network and stays there undetected for a long period of time”*

✓ APT attacks target organizations in sectors with high-value information, such as **military networks**, **national defense**, manufacturing and the financial industry

https://www.damballa.com/

APTs can best be summarized by their named requirements:

-Advanced

Criminal operators behind the threat utilize the full spectrum of computer intrusion technologies and techniques. While individual components of the attack may not be classed as particularly “advanced” (e.g. malware components generated from commonly available Do-It-Yourself (DIY) construction kits, or the use of easily procured exploit materials), their operators can typically access and develop more advanced tools as required. They combine multiple attack methodologies and tools in order to reach and compromise their target.

APTs can best be summarized by their named requirements:

-Persistent

Criminal operators give priority to a specific task, rather than opportunistically seeking immediate financial gain. This distinction implies that the attackers are guided by external entities. The attack is conducted through continuous monitoring and interaction in order to achieve the defined objectives. It does not mean a barrage of constant attacks and malware updates. In fact, a “low-and-slow” approach is usually more successful.

APTs can best be summarized by their named requirements:

-Threat

means that there is a level of coordinated human involvement in the attack, rather than a mindless and automated piece of code. The criminal operators have a specific objective and are skilled, motivated, organized and well funded.

APTs are different from other targeted attacks in the following ways:

-Customized attacks

In addition to more common attack methods, APTs often use highly customized tools and intrusion techniques, developed specifically for the campaign. These tools include zero-day vulnerability exploits, viruses, worms, and rootkits. In addition, APTs often launch multiple threats or "kill chains" simultaneously to breach their targets and ensure ongoing access to targeted systems, sometimes including a "sacrificial" threat to trick the target into thinking the attack has been successfully repelled.

APTs are different from other targeted attacks in the following ways:

-Low and slow

APT attacks occur over long periods of time during which the attackers move slowly and quietly to avoid detection. In contrast to the “smash and grab” tactics of many targeted attacks launched by more typical cybercriminals, the goal of the APT is to stay undetected by moving “low and slow” with continuous monitoring and interaction until the attackers achieve their defined objectives.

APTs are different from other targeted attacks in the following ways:

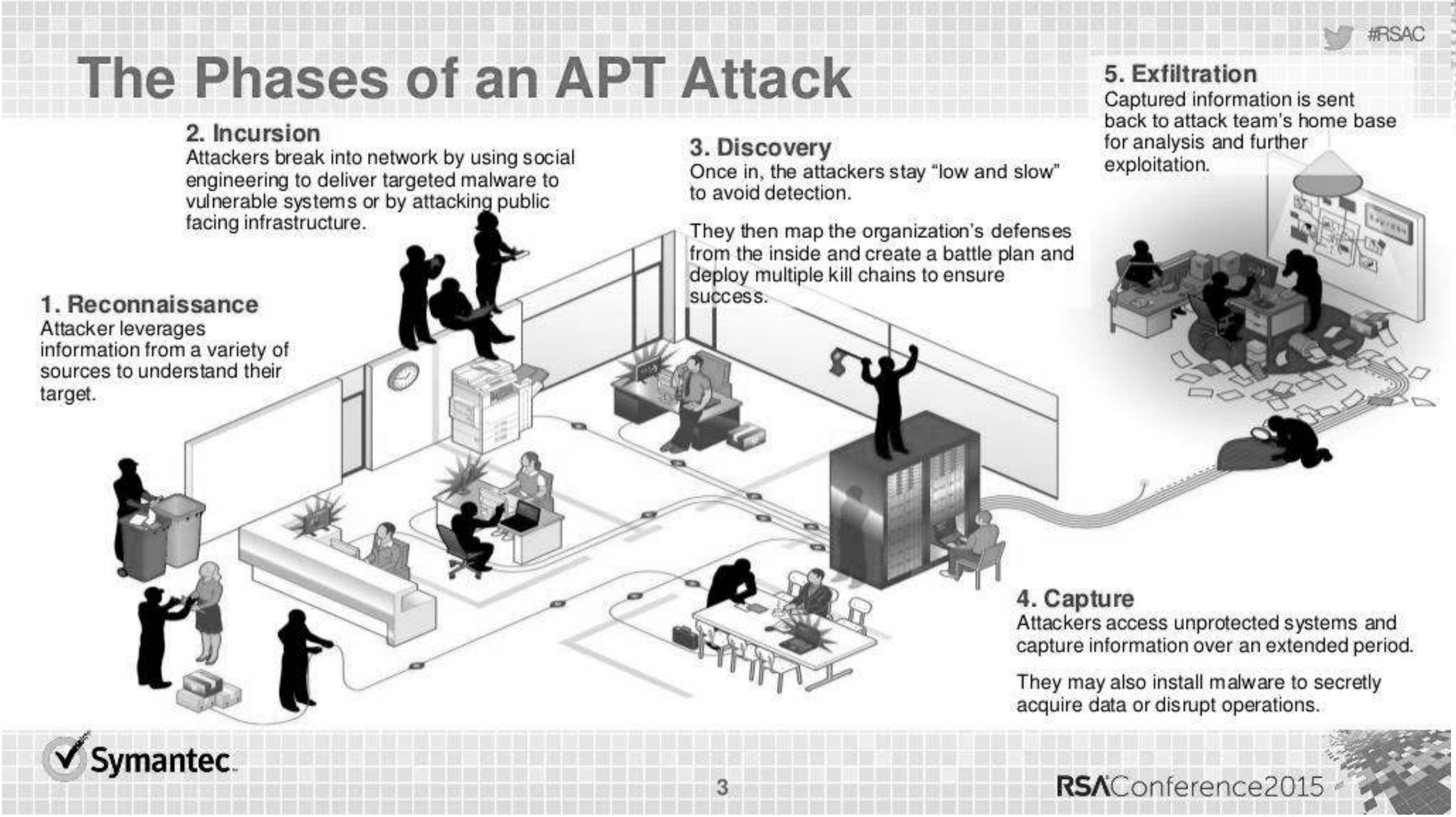
-Higher aspirations

Unlike the fast-money schemes, APTs are designed to satisfy the requirements of international espionage and/or sabotage, usually involving covert state actors. The objective may include military, political, or economic intelligence gathering, confidential data or trade secret threat, disruption of operations, or even destruction of equipment. The groups behind APTs are well funded and staffed; they may operate with the support of military or state intelligence.

APTs are different from other targeted attacks in the following ways:

-Specific targets

Widely reported APT attacks have been launched at government agencies and facilities, defense contractors, and manufacturers of products that are highly competitive on global markets. In addition, APTs may attack vendor or partner organizations that do business with their primary targets. Also ordinary companies with valuable technology or intellectual property and organizations that maintain and operate vital national infrastructure are also likely targets.



#RSAC

The Phases of an APT Attack

1. Reconnaissance
Attacker leverages information from a variety of sources to understand their target.

2. Incursion
Attackers break into network by using social engineering to deliver targeted malware to vulnerable systems or by attacking public facing infrastructure.

3. Discovery
Once in, the attackers stay "low and slow" to avoid detection.

They then map the organization's defenses from the inside and create a battle plan and deploy multiple kill chains to ensure success.

4. Capture
Attackers access unprotected systems and capture information over an extended period.

They may also install malware to secretly acquire data or disrupt operations.

5. Exfiltration
Captured information is sent back to attack team's home base for analysis and further exploitation.

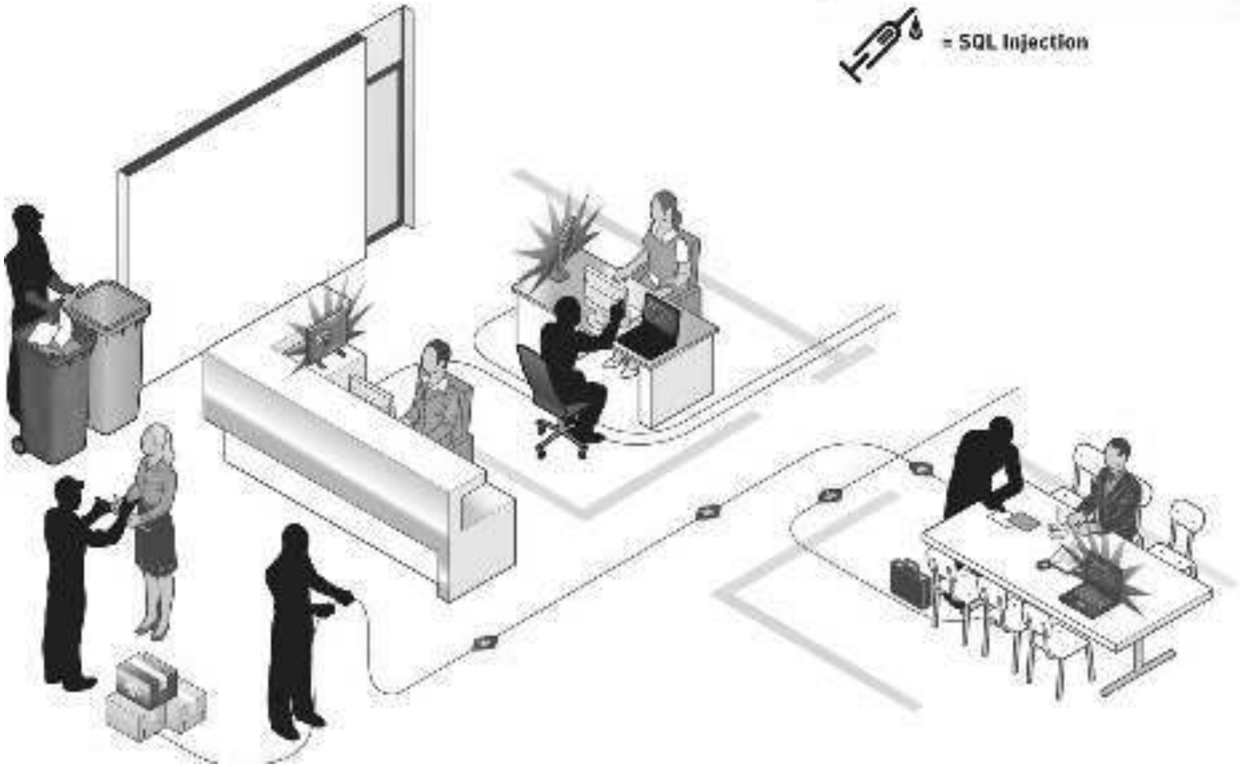
How do APT attacks work?

1. INCURSION

Attackers break into network by using social engineering to deliver targeted malware to vulnerable systems and people

ATTACK METHODS

- = Social Engineering
- = Zero-Day Vulnerability
- = SQL Injection



Bots & Botnets

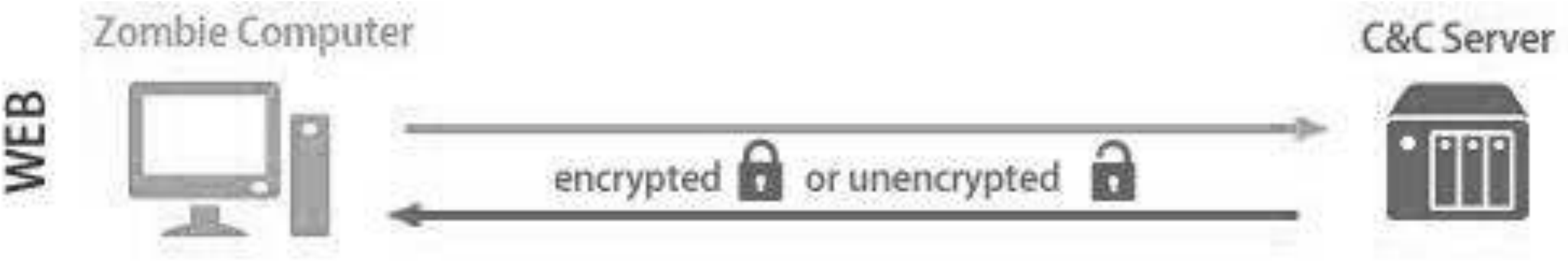
Bots are one of the most sophisticated and popular types of cybercrime today.

They allow hackers to take control of many computers at a time, and turn them into "zombie" computers, which operate as part of a powerful "botnet".

Botnets employ evolving techniques to obfuscate the specific host involved in their phishing schemes, malware delivery or other criminal enterprises, like **money mule recruitment sites, illicit online pharmacies, extreme or illegal adult content sites, malicious browser exploit sites and web traps for distributing virus.**

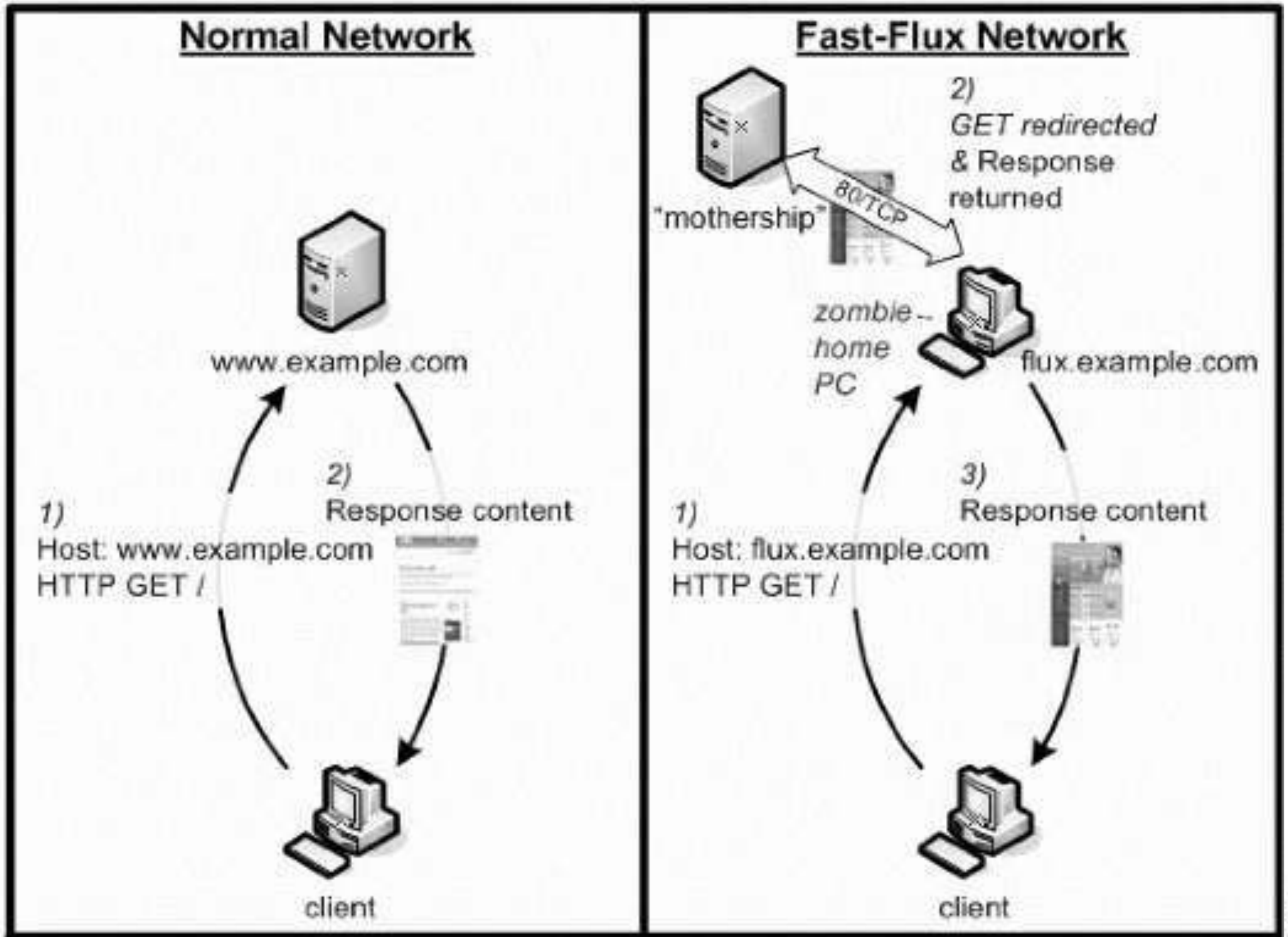
What is it about?

- ✓ One of the biggest challenges for botnet owners is the protection of Command-and-Control traffic. C&C traffic is required to give orders to the "zombies", the infected computers that are part of the botnets.
- ✓ Generally, up to now, two approaches existed for C&C traffic:
  - Either a central control server is put somewhere on the Internet or
  - Peer-to-Peer-networks (P2P) are built up to ensure the chain of commands.



IP-Flux

- ✓ refers to the constant changing of IP address information (e.g. 192.168.1.1) related to a particular, fully qualified domain name (e.g. mypc.atl.damballa.com).
- ✓ Botnet operators abuse this ability to change IP address information associated with a host name by linking multiple IP addresses with a specific host name and rapidly changing the linked addresses.
- ✓ These IPs are interchanged too fast, with a very small Time-To-Live (TTL) for each partial DNS Resource Record.
- ✓ In this way a domain name can change its corresponding IP address very often (e.g. every 3 minutes).
- ✓ This rapid changing aspect is referred to as **“Fast-Flux”**.



Normal Network

www.example.com

1) Host: www.example.com HTTP GET /

2) Response content

client

Fast-Flux Network

"mothership"

80/TCP

2) GET redirected & Response returned

zombie home PC

flux.example.com

1) Host: flux.example.com HTTP GET /

3) Response content

client

Web Request Comparison

Fast-Flux Botnets

- ✓ **Single-flux**
  - is characterized by having multiple IP addresses associated with a domain name. These IP addresses are registered and de-registered rapidly – using a combination of round-robin allocation and very short TTL values against a particular DNS Resource Record. *DNS* ***A*** *records that change quickly*.
- ✓ **Double-flux**
  - not only fluxes the IP addresses associated with the fully-qualified domain name (FQDN), but also fluxes the IP addresses of the DNS servers (e.g., NS records) that are in turn used to lookup the IP addresses of the FQDN. *DNS* ***A and NS*** *records change quickly.*

Blind Proxy Redirection (BPR)

- ✓ Redirection disrupts attempts to track down and mitigate fast-flux service network nodes.
- ✓ What happens is the large pool of rotating IP addresses are not the final destination of the request for the content (or other network service).
- ✓ Instead, compromised front end systems are merely deployed as redirectors that funnel requests and data to and from other backend servers, which actually serve the content. Essentially the domain names and URLs for advertised content no longer resolve to the IP address of a specific server, but instead fluctuate amongst many front end redirectors or proxies, which then in turn forward content to another group of backend servers.

Domain Flux

- ✓ is effectively the inverse of IP flux and refers to the constant changing and allocation of multiple FQDN’s to a single IP address or C&C infrastructure.
- ✓ **Domain Wildcarding**
  - abuses native DNS functionality to wildcard (e.g., *) a higher domain such that all FQDN’s point to the same IP address. For example, ***.damb.com** could encapsulate both **mypc.atl.damb.com** and **server.damb.com**. This technique is most commonly associated with spam or phishing botnets – whereby the wildcarded information that appears random (e.g. “asdk” of asdk.atl.damb) is used by the botmasters to uniquely identify a victim, track success using various delivery techniques, and bypass anti-spam technologies.

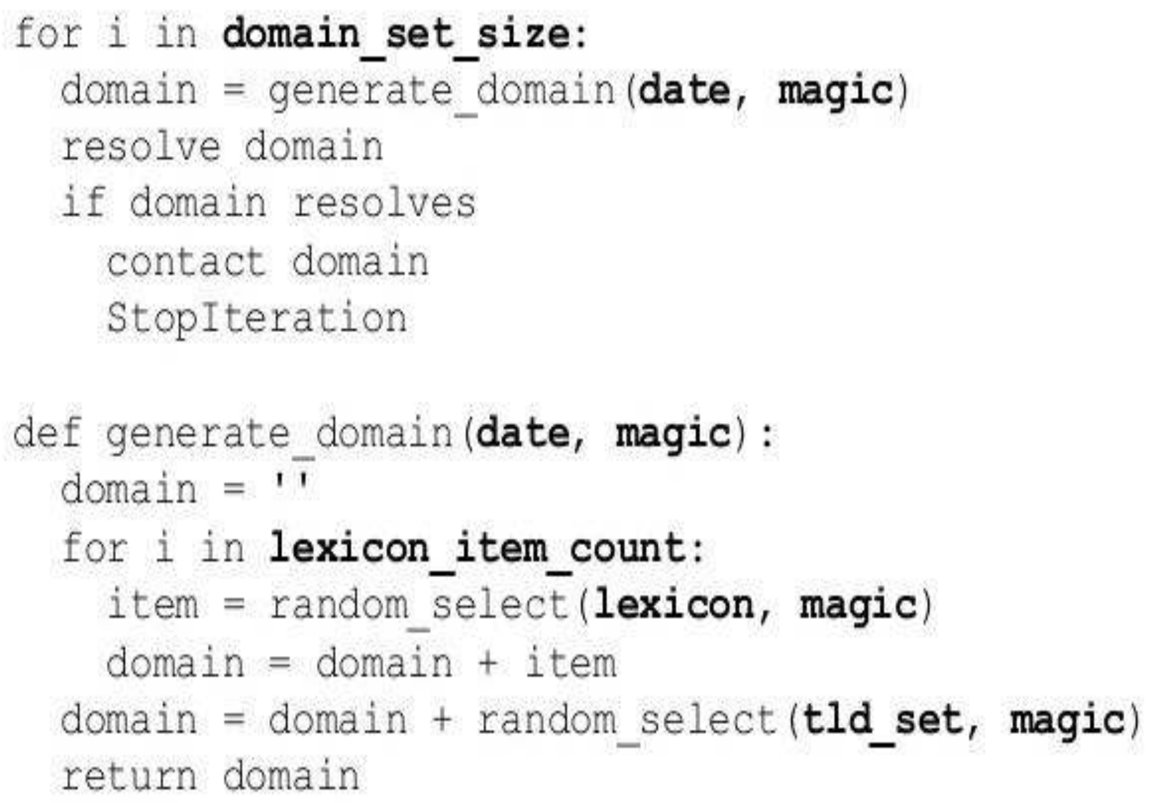
Domain Generation Algorithm (DGA)

- ✓ Bot agents create a dynamic list of multiple FQDN’s that can be used as rendezvous points with their C&C servers.
- ✓ The large number of potential rendezvous points makes it difficult for law enforcement to effectively shut down botnets since infected computers will attempt to contact some of these domain names every day to receive updates or commands.
- ✓ By using public-key cryptography, it is unfeasible for law enforcement and other actors to mimic commands from the malware controllers as some worms will automatically reject any updates not signed by the malware controllers.

Domain Generation Algorithm (DGA)

- ✓ For example,
  - an infected computer could create thousands of domain names such as: **www.gi9bfb4er2ig4fws8h.ir** and would attempt to contact a portion of these with the purpose of receiving an update or commands.
- ✓ Embedding the DGA instead of a list of previously-generated (by the C&C servers) domains in the unobfuscated binary of the malware protects against a strings dump that could be fed into a network blacklisting appliance preemptively to attempt to restrict outbound communication from infected hosts within an enterprise.

Generalized DGA pseudo code…



```
for i in domain_set_size:
  domain = generate_domain(date, magic)
  resolve domain
  if domain resolves
    contact domain
    StopIteration

def generate_domain(date, magic):
  domain = ''
  for i in lexicon_item_count:
    item = random_select(lexicon, magic)
    domain = domain + item
  domain = domain + random_select(tld_set, magic)
  return domain
```

Example DGA Output
gi9bfb4er2ig4fws8h.ir
vfxlsatformalisticirekb.com
rd0ee55073a3776810962c.ws
croialotvvnfliyjmvt.ru
yxjsibe5ugmmj.in
osghqr87dmlyhh.net
eas1ebr1ainj4obmarket.com

The next step made – using the Tor network

- ✓ Tor is generally known as web anonymization service for end users, but Tor offers more than that: “Tor makes it possible for users to hide their locations while offering various kinds of services, such as web publishing or an instant messaging server.”
- ✓ Tor directs Internet traffic through a free, worldwide, volunteer network consisting of more than seven thousand relays to conceal a user's location and usage from anyone conducting network surveillance or traffic analysis.
- ✓ Using Tor makes it more difficult for Internet activity to be traced back to the user.
- ✓ In this particular case, the creators of the malware decided to build an IRC server as hidden service.

Botnets over the Tor network

- ✓ gains the botmaster several advantages:
  - The server is anonymous and thus cannot point to the botnet owners’ identity.
  - The server cannot be taken down easily.
  - The traffic is encrypted by Tor, so it can’t be blocked by Intrusion Detection Systems.
  - Tor traffic usually cannot be blocked altogether, because there are also legit use cases for Tor.
  - The bot creator does not necessarily have to generate a custom protocol, but can use the known and reliable IRC protocol.

Tor vs HTTPS

- ✓ Typically, Tor uses TLS over TCP as its transport protocol. The well known TLS port for Tor traffic is 443, which is used by the HTTPS, so that the interpretation of a session exclusively with the determination of the door cannot constitute a reliable method.
- ✓ A successful method for detecting Tor traffic is the statistical analysis and the identification of the Secure Sockets Layer (SSL) protocol differences.
- ✓ The SSL protocol uses a combination of public and symmetric key encryption. Each SSL connection always starts with the exchange of messages by the server and the client until the secure connection is established (handshake).

Tor vs HTTPS

- ✓ The handshake allows the server to prove its identity to the client by using public-key encryption techniques and then allows the client and the server to cooperate in the creation of a symmetric key to be used to quickly encrypt and decrypt data exchanged between them.
- ✓ Optionally, the handshake also allows the client to prove its identity to the server.
- ✓ Given that each Tor client creates self-signed SSL, using a random domain name that changes around every 30 minutes, a statistical analysis of the network traffic based on the specific SSL characteristics can identify the Tor sessions, in a network full of HTTPS traffic.

A traditional perimeter security must be extended to include the following functions:

- ✓ **Advanced detection**: Analysis of traffic and application patterns if an attack is in progress.
- ✓ **Total containment**: Once an attack is detected, it must be ensured that this cannot spread further.
- ✓ **Threat identification**: The deployment of a "key-learning tool" which learns with every attack and improves long-term protection.
- ✓ **Advanced threat mitigation**: This includes the targeted removal of malware, the reorganization of hazardous systems and restore after an attack.

Real-time Computational Intelligence Protection Framework Against APTs:

- ✓ an innovative, fast and accurate real-time Computational Intelligence Protection Framework against APTs (CIPFaAPTs), that performs analysis of network flow, in order to perform malware traffic analysis, network traffic classification and fast-flux botnets localization,
- ✓ this is achieved by employing **Online Sequential Extreme Learning Machines** with **Gaussian Radial Basis Function** kernel (**OS-ELM GRBFk**),
- ✓ offers high learning speed, ease of implementation and minimal human intervention.

Datasets:

- ✓ 4 datasets were constructed and used for testing CIPFaAPT:
  - ✓ **Domain Generation Algorithms dataset** (DGA_dataset) 5 features + class (legit or malicious), containing 131,374 patterns (100,000 URLs they were chosen randomly from the database with the 1 million most popular domain names of Alexa and 16,374 malicious URLs from the updated list of the Black Hole DNS database and 15,000 malicious URLs they were created based on the timestamp DGA algorithm).

Datasets:

- ✓ 4 datasets were constructed and used for testing CIPFaAPT:
  - ✓ **Malware Traffic Analysis dataset** (MTA_dataset) 32 features + class (benign or malware). The MTA_dataset containing 73469 patterns (37127 benign samples they were chosen from the Pcaps from National CyberWatch Mid-Atlantic Collegiate Cyber Defense Competition and 36342 malicious samples they were chosen from http://malware-traffic-analysis.net/).

Datasets:

- ✓ 4 datasets were constructed and used for testing CIPFaAPT:
  - ✓ **Network Traffic Classification dataset** (NTC_dataset) 22 features + 12 classes (TELNET, FTP, HTTP, HTTPS, DNS, Lime, Local Forwarding, Remote Forwarding, SCP, SFTP, x11, Shell), containing 137050 patterns they were chosen from the Pcaps from Information Technology Operations Center (ITOC), US Military Academy.

ITOC
UNITED STATES MILITARY ACADEMY
WEST POINT.
Information Technology and Operations Center

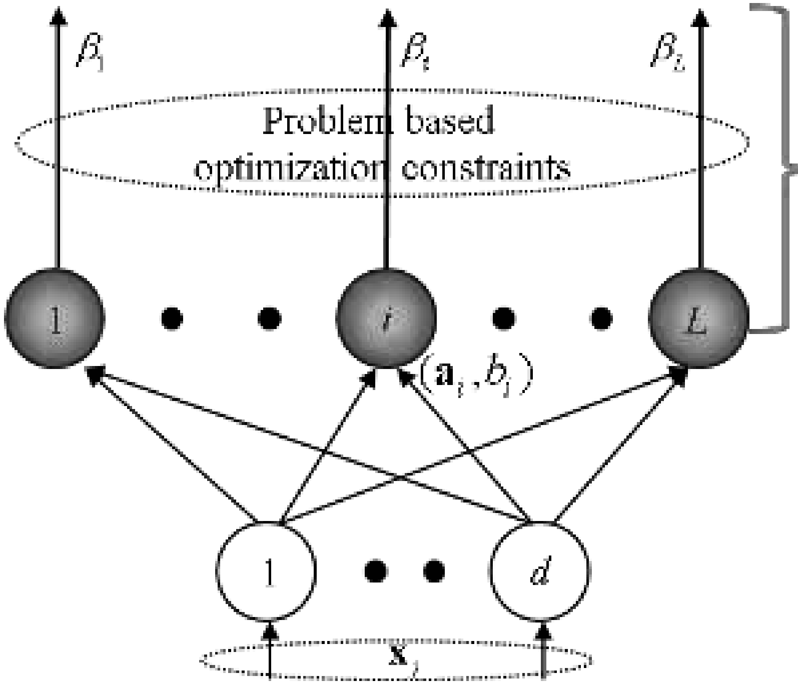
Datasets:

- ✓ 4 datasets were constructed and used for testing CIPFaAPT:
  - ✓ **Tor-Traffic Identification dataset** (TTI_dataset) 45 features and 2 classes (Tor or HTTPS), containing 217,483 patterns they were chosen from the Pcaps from FOI's Information Warfare Lab of Swedish Defense Research Agency.

FOI
Swedish Defense
Research Agency

Methodologies:

Extreme Learning Machines (ELM)



Feature learning
Clustering
Regression
Classification

L Random Hidden Neurons (which need not be algebraic sum based) or other ELM feature mappings. Different type of output functions could be used in different neurons:

$$h_i(\mathbf{x}) = G_i(\mathbf{a}_i, b_i, \mathbf{x})$$

d Input Nodes

Methodologies:

Online Sequential ELM with Gaussian RBF kernel (OS-ELM GRBFk)

✓ It is a versatile sequential learning algorithm in the following sense:

- the training observations are sequentially (one-by-one or chunk-by-chunk with varying or fixed chunk length) presented to the learning algorithm,
- at any time, only the newly arrived single or chunk of observations (instead of the entire past data) are seen and learned,
- a single or a chunk of training observations is discarded as soon as the learning procedure for that particular (single or chunk of) observation(s) is completed,

Methodologies:

Online Sequential ELM with Gaussian RBF kernel (OS-ELM GRBFk)

✓ It is a versatile sequential learning algorithm in the following sense:

- the learning algorithm has no prior knowledge as to how many training observations will be presented.
- nodes, the centers and widths of the nodes are randomly generated and fixed and then, based on this, the output weights are analytically determined,
- unlike other sequential learning algorithms which have many control parameters to be tuned, OS-ELM with GRBFk only requires the number of hidden nodes to be specified.

Results of DGA_dataset

✓ The performance comparisons of algorithms:

Classifier	**Properties**		**Classification Accuracy & Performance Metrics**					
	Activation Function	Learning Mode	ACC	RMSE	Precision	Recall	F-Score	ROC Area
ELM	RBF	Batch	92.17%	0.1877	0.920%	0.921	0.921%	0.975
ELM	Sigmoid	Batch	91.35%	0.2031	0.914%	0.914	0.914%	0.960
OS-ELM	RBF	1 by 1	92.89%	0.1804	0.930%	0.929	0.929%	0.978
OS-ELM	Sigmoid	20 by 20	93.13%	0.1726	0.932%	0.932	0.932%	0.982
OS-ELM	**RBF**	**20 by 20**	**93.97%**	**0.1711**	**0.940%**	**0.940**	**0.940%**	**0.985**
OS-ELM	Sigmoid	1 by 1	91.92%	0.2012	0.919%	0.919	0.920%	0.963

Results of MTA_dataset

✓ The performance comparisons of algorithms:

Classifier	**Properties**		**Classification Accuracy & Performance Metrics**					
	Activation Function	Learning Mode	ACC	RMSE	Precision	Recall	F-Score	ROC Area
ELM	RBF	Batch	96.71%	0.1421	0.967%	0.967	0.967%	0.980
ELM	Sigmoid	Batch	96.64%	0.1432	0,967%	0.966	0.966%	0.979
OS-ELM	RBF	1 by 1	98.28%	0.1342	0.982%	0.983	0.983%	0.985
OS-ELM	Sigmoid	20 by 20	96.99%	0.1426	0.970%	0.970	0.970%	0.970
OS-ELM	**RBF**	**20 by 20**	**98.34%**	**0.1331**	**0.983%**	**0.984**	**0.983%**	**0.990**
OS-ELM	Sigmoid	1 by 1	96.81%	0.1429	0,969%	0.969	0.970%	0.969

Results of NTC_dataset

✓ The performance comparisons of algorithms:

Classifier	**Properties**		**Classification Accuracy & Performance Metrics**					
	Activation Function	Learning Mode	ACC	RMSE	Precision	Recall	F-Score	ROC Area
ELM	RBF	Batch	99.15%	0.1027	0.991%	0.991	0.992%	0.991
ELM	Sigmoid	Batch	99.11%	0.1030	0,991%	0.990	0.990%	0.990
OS-ELM	RBF	1 by 1	99.51%	0.1006	0.995%	0.995	0.995%	0.995
OS-ELM	Sigmoid	20 by 20	99.68%	0.0990	0.996%	0.997	0.996%	0.996
OS-ELM	**RBF**	**20 by 20**	**99.72%**	**0.0982**	**0.998%**	**0.997**	**0.997%**	**0.997**
OS-ELM	Sigmoid	1 by 1	99.44%	0.1016	0.994%	0.994	0.994%	0.994

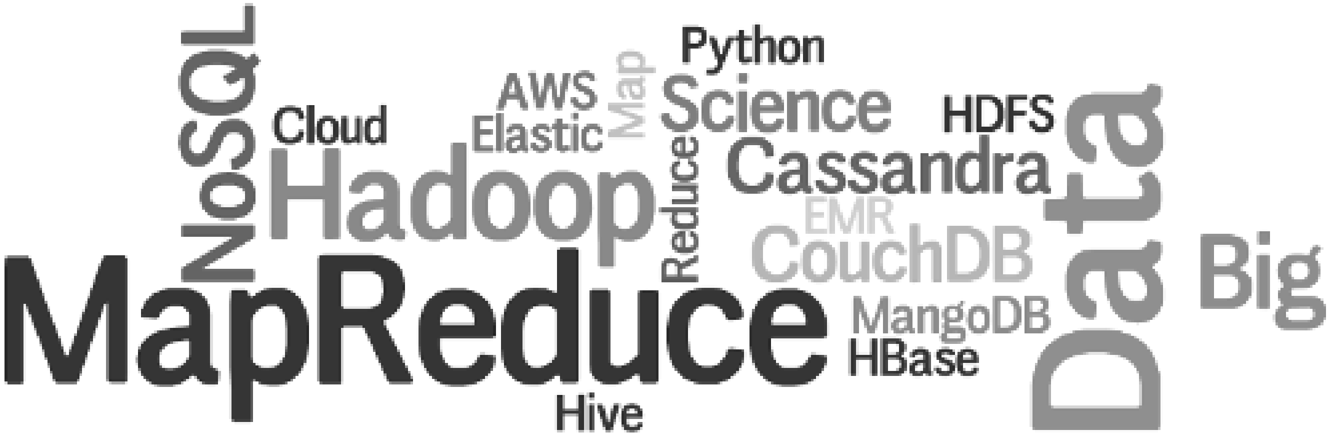
Results of TTI_dataset

- ✓ The performance comparisons of algorithms:

Classifier	**Properties**		**Classification Accuracy & Performance Metrics**					
	Activation Function	Learning Mode	ACC	RMSE	Precision	Recall	F-Score	ROC Area
ELM	RBF	Batch	94.19%	0.1561	0.942%	0.942	0.942%	0.942
ELM	Sigmoid	Batch	94.10%	0.1570	0.941%	0.941	0.941%	0.941
OS-ELM	RBF	1 by 1	94.31%	0.1537	0.943%	0.943	0.943%	0.970
OS-ELM	Sigmoid	20 by 20	94.24%	0.1543	0.942%	0.943	0.943%	0.965
OS-ELM	**RBF**	**20 by 20**	**94.39%**	**0.1521**	**0.944%**	**0.944**	**0.944%**	**0.970**
OS-ELM	Sigmoid	1 by 1	94.28%	0.1539	0.943%	0.943	0.943%	0.943

Future Directions

- ✓ feature minimization using **Principal Component Analysis** (**PCA**) or other existing approaches,
- ✓ additional computational intelligence methods such as **Spiking Neural Networks** could be explored and compared on the same security task and
- ✓ **big data** tools.



Conclusions

- ✓ An innovative Real-time Computational Intelligence Protection Framework Against Advanced Persistent Threats has been introduced.
- ✓ It is a next generation security platform that uses sophisticated analytics to monitor, track and classify risk across critical network infrastructures in order to identify APT.
- ✓ It performs classification by using an **Online Sequential ELM with Gaussian RBF kernel**, a very fast approach with high accuracy and generalization with minimum computational power and resources.

References

[1] Chi Cheng, Wee Peng Tay, Guang-Bin Huang, Extreme learning machines for intrusion detection, Neural Networks (IJCNN), International Joint Conference, DOI: 10.1109/IJCNN.2012.6252449, 2012.

[2] Erik Cambria, Guang-Bin Huang, Extreme Learning Machines, IEEE InTeLLIGenT SYSTemS, 541-1672/13, 2013.

[3] Alshammari, Riyad; Nur Zincir-Heywood, A., A flow based approach for SSH traffic detection, Systems, Man and Cy-bernetics, 2007. ISIC. IEEE International Conference on , pp.296-301, 7-10 Oct, 2007.

[4] Holz T., C. Gorecki, K. Rieck, and F. Freiling, Measuring and detecting fast-flux service networks, in NDSS ’08: Proceedings of the Network & Distributed System Security Symposium, 2008.

[5] Nan-Ying Liang, Guang-Bin Huang, P. Saratchandran, and N. Sundararajan, A Fast and Accurate Online Sequential Learning Algorithm for Feedforward Networks, IEEE Transactions on Neural Networks, Vol. 17, No. 6, 2006.

Thanks

kdemertz@fmenr.duth.gr
http://utopia.duth.gr/~kdemertz/