Practical Detection of Credential Spearphishing at LBNL

Vern Paxson
UC Berkeley / Corelight, Inc.

Grant Ho (UCB), Aashish Sharma (LBNL), Mobin Javed (LUMS), David Wagner (UCB)

May 22, 2018
Spear Phishing

Targeted email that tricks victim into giving attacker privileged capabilities
To: vern@ee.lbl.gov
Subject: RE: Russian spear phishing attack against .mil and .gov employees
From: jeffreyc@cia.gov
Date: Wed, 10 Feb 2010 19:51:47 +0100

Russian spear phishing attack against .mil and .gov employees

A "relatively large" number of U.S. government and military employees are being taken in by a spear phishing attack which delivers a variant of the Zeus trojan. The email address is spoofed to appear to be from the NSA or IntelLink concerning a report by the National Intelligence Council named the "2020 Project". It's purpose is to collect passwords and obtain remote access to the infected hosts.

Security Update for Windows 2000/XP/Vista/7 (KB823988)

About this download: A security issue has been identified that could allow an attacker to remotely compromise a computer running Microsoft Windows and gain complete control over it. You can help protect your computer by installing this update from Microsoft. After you install this item, you may have to restart your computer.

Download:

http://mv.net.md/update/update.zip

or

http://www.sendspace.com/file/xwc1pi

Jeffrey Carr is the CEO of GreyLogic, the Founder and Principal Investigator of Project Grey Goose, and the author of "Inside Cyber Warfare".
jeffreyc@greylogic.us
To: vern@ee.lbl.gov  
Subject: RE: Russian spear phishing attack against .mil and .gov employees  
From: jeffreyc@cia.gov  
Date: Wed, 10 Feb 2010 19:51:47 +0100

Russian spear phishing attack against .mil and .gov employees

A "relatively large" number of U.S. government and military employees are being taken in by a spear phishing attack which delivers a variant of the Zeus trojan. The email address is spoofed to appear to be from the NSA or IntelLink concerning a report by the National Intelligence Council named the "2020 Project". Its purpose is to collect passwords and obtain remote access to the infected hosts.

Security Update for Windows 2000/XP/Vista/7 (KB823988)

About this download: A security issue has been identified that could allow an attacker to remotely compromise a computer running Microsoft Windows and gain complete control over it. You can help protect your computer by installing this update from Microsoft. After you install this item, you may have to restart your computer.

Download:

http://mv.net.md/update/update.zip

or

http://www.sendspace.com/file/xwc1pi

Jeffrey Carr is the CEO of GreyLogic, the Founder and Principal Investigator of Project Grey Goose, and the author of "Inside Cyber Warfare".  
jeffreyc@greylogic.us

Yep, this email was itself a spearphish
Hello,

The University is having a salary increment program again this year with an average of 2.5%.

The Human Resources department evaluated you for a raise on your next paycheck.

Click below to confirm and access your salary revision documents:

USERNAME: 
PASSWORD: 

If you have any questions, please contact Human Resources at hrinfo@caltech.edu.
Our Work

*Practical* system for an enterprise’s security team for detecting spearphishing of *login credentials*

1. Extremely low FP burden (Goal: < *minutes per day*)

2. Raises bar & detects many attacks; *not* silver bullet
Anchoring the Research

Worked with (anonymized) datasets:
  • SMTP header information (From and RCPT-TO) (but not content)
  • URLs in emails
  • Network traffic logs
  • LDAP logs
Key Challenges

1. Small set of labeled attack data
   • < 10 known successful credential spearphishing attacks

2. Base rate
   • 372 million emails over 4 years (Mar 2013 – Jan 2017)
   • Even detector w/ 99.9% accuracy = 372,000 alerts
Structure-Driven Features
Spearphishing Attack Taxonomy

Successful spearphishing attacks have 2 necessary stages:

1. The *Lure*
   - Successful attacks *lure* / convince victim to perform an action

2. The *Exploit*
   - Successful attacks execute some *exploit* on behalf of the attacker
   - Malware, revealing credentials, wiring money to “corporate partner”
Spearphishing Attack Taxonomy

Successful spearphishing attacks have 2 necessary stages:

1. The *Lure*  
   • Successful attacks *lure*/convince victim to perform an action

2. The *Exploit*  
   • Successful attacks execute some *exploit* on behalf of the attacker  
   • Malware, *revealing credentials*, wiring money to “corporate partner”
Modern Credential Spearphishing: The Lure

Lure

1. Attacker sends catchy email under trusted/authoritative identity

From: "Berkeley IT Staff"
<security@berkeley.net>
Modern Credential Spearphishing: The Exploit

Exploit
1. Victim *clicks on embedded link*
2. Victim arrives at phishing website & *submits credentials*
## Types of Lures

<table>
<thead>
<tr>
<th>Type of Impersonation</th>
<th>Forge Name</th>
<th>Forge Email</th>
<th>Real Life Example</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: Lateral Attacker (Stolen Credentials). No need to spoof, given account ownership.
# Types of Lures

<table>
<thead>
<tr>
<th>Type of Impersonation</th>
<th>Forge Name</th>
<th>Forge Email</th>
<th>Real Life Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address Spoofer</td>
<td>May or may not</td>
<td>YES</td>
<td><img src="image" alt="Example Email" /></td>
</tr>
</tbody>
</table>

Note: The example email shows how an Address Spoofer might be used in a real-life scenario.
# Types of Lures

<table>
<thead>
<tr>
<th>Type of Impersonation</th>
<th>Forge Name</th>
<th>Forge Email</th>
<th>Real Life Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address Spoofer</td>
<td>May or may not</td>
<td>YES</td>
<td><img src="image1.png" alt="Email Example 1" /></td>
</tr>
<tr>
<td>Historically New Attacker</td>
<td>Unseen Name</td>
<td>Unseen Email (@lbl.gov = forged)</td>
<td><img src="image2.png" alt="Email Example 2" /></td>
</tr>
</tbody>
</table>

**Note:**
- The table provides examples of different types of lures, including address spoofer and historically new attacker.
- REAL LIFE EXAMPLE:
  - **Address Spoofer**
    - Name: May or may not
    - Email: YES
    - Real Life Example:
      ```
      Date: Fri, 14 Aug 2015 12:04:00 -0500 (CDT)
      From: "A Alivisatos" <aalivisatos@lbl.gov>
      To: XXXXXX@lbl.gov
      Subject: Good Morning
      Reply-To: aalivisatoslbl@mail.com
      Send me the balance on all our accounts as of today's date.
      Thanks.
      ```
  - **Historically New Attacker**
    - Name: Unseen
    - Email: Unseen Email (@lbl.gov = forged)
    - Real Life Example:
      ```
      Date: Tue, 08 Nov 2016 17:38:28 +0000
      From: Computer Maintenance <compmaint@lbl.gov>
      To: afXXXXXXXXXh@lbl.gov
      Subject: Urgent: Email reactivation
      ```
## Types of Lures

<table>
<thead>
<tr>
<th>Type of Impersonation</th>
<th>Forge Name</th>
<th>Forge Email</th>
<th>Real Life Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address Spoofer</td>
<td>May or may not</td>
<td>YES</td>
<td><img src="" alt="Example 1" /></td>
</tr>
<tr>
<td>Historically New Attacker</td>
<td>Unseen Name</td>
<td>Unseen Email (@lbl.gov = forged)</td>
<td><img src="" alt="Example 2" /></td>
</tr>
<tr>
<td>Name Spoofer</td>
<td>YES</td>
<td>YES</td>
<td><img src="" alt="Example 3" /></td>
</tr>
</tbody>
</table>

### Note

- In the example email, the email address `alvisatos@lbl.gov` is forged to `aalvisatosl@mail.com` which is an example of name spoofing.
## Types of Lures

<table>
<thead>
<tr>
<th>Type of Impersonation</th>
<th>Forge Name</th>
<th>Forge Email</th>
<th>Real Life Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address Spoofer</td>
<td>May or may not YES</td>
<td></td>
<td><img src="image1" alt="Email Content" /></td>
</tr>
<tr>
<td>Historically New Attacker</td>
<td>Unseen Name Unseen Email (@lbl.gov = forged)</td>
<td></td>
<td><img src="image2" alt="Email Content" /></td>
</tr>
<tr>
<td>Name Spoofer</td>
<td>YES</td>
<td>YES</td>
<td><img src="image3" alt="Email Content" /></td>
</tr>
<tr>
<td>Lateral Attacker (Stolen Credentials)</td>
<td>No need to spoof, given account ownership No need to spoof, given account ownership</td>
<td></td>
<td><img src="image4" alt="Email Content" /></td>
</tr>
</tbody>
</table>

Notes:
- ![Note Image](image5)
### Types of Lures

<table>
<thead>
<tr>
<th>Type of Impersonation</th>
<th>Forge Name</th>
<th>Forge Email</th>
<th>Real Life Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address Spoofer</td>
<td>May or may not</td>
<td>YES</td>
<td><img src="image" alt="Example Email" /></td>
</tr>
<tr>
<td>Historically New Attacker</td>
<td>Unseen Name</td>
<td>Unseen Email (@lbl.gov = forged)</td>
<td><img src="image" alt="Example Email" /></td>
</tr>
<tr>
<td>Name Spoofer</td>
<td>YES</td>
<td>YES</td>
<td><img src="image" alt="Example Email" /></td>
</tr>
<tr>
<td>Lateral Attacker (Stolen</td>
<td>No need to spoof, given account ownership</td>
<td>No need to spoof, given account ownership</td>
<td><img src="image" alt="Example Email" /></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note</strong>: Each lure type has its own associated features.</td>
</tr>
</tbody>
</table>

Each lure type has its own associated features.
Lure Features: Suspicious Sender Present

• E.g., lateral spearphishing lure: attacker compromises trusted entity’s account

• Feature intuition: email = suspicious if employee sent it during an unusual login session

• **Lure** features for **lateral spearphishing**:  
  • email sent in session where sender logged in w/ new IP addr?  
  • # prior logins by sender from geolocated city of login IP addr  
  • # of other employees who’ve also logged in from that city
Exploit Features: Suspicious Action Occurred

• Winnow pool of candidate alerts to:
  Emails where recipient clicked on embedded URL (a *click-in-email* action)

• **Exploit** features: URL’s *fully-qualified domain* (hostname) is unusual:
  • # of prior visits to FQDN across all enterprise’s network traffic
  • # of days between 1st employee visit to FQDN & current email’s arrival
Using Features for Detection
How do we leverage our features?

• How do we decide if lure + exploit features indicate a possible attack?

Approach 1: Manual rules

• Problems: soundly choosing thresholds; generalizability
How do we leverage our features?

- How do we decide if lure + exploit features indicate a possible attack?

**Approach 1: Manual rules**
- Problems: soundly choosing thresholds; generalizability

**Approach 2: Supervised ML**
- Problems: tiny # of labeled attacks; base rate
How do we leverage our features?

- How do we decide if lure + exploit features indicate a possible attack?

**Approach 1: Manual rules**
- Problems: soundly choosing thresholds; generalizability

**Approach 2: Supervised ML**
- Problems: tiny # of labeled attacks; base rate

**Approach 3: Unsupervised learning** *(anomaly detection)*
- Clustering/Distance Based: kNN
- Density-based: KDE, GMM
- (Many others ...)
Classical Anomaly Detection: Limitations

Three general problems:
Classical Anomaly Detection: Limitations

Three general problems:

1. Parametric and/or hyperparameter tuning
Classical Anomaly Detection: Limitations

Three general problems:

1. Parametric and/or hyperparameter tuning

2. Direction-agnostic (standard dev of +3 just as anomalous as -3)

Feature:
# prior logins by current employee from city of new IP addr
Classical Anomaly Detection: Limitations

Three general problems:

1. Parametric and/or hyperparameter tuning
2. Direction-agnostic (standard dev of +3 just as anomalous as -3)
3. Alert if anomalous in only one dimension
Classical Anomaly Detection: Limitations

Three general problems:

1. Parametric and/or hyperparameter tuning
2. Direction-agnostic (standard dev of +3 just as anomalous as -3)
3. Alert if anomalous in only one dimension
Classical Anomaly Detection: Limitations

Three general problems:

1. Parametric and/or hyperparameter tuning
2. Direction-agnostic (standard dev of +3 just as anomalous as -3)
3. Alert if anomalous in only one dimension

**DAS**: *simple*, new method that overcomes these 3 problems
Classical Anomaly Detection: Limitations

Three general problems:

1. Parametric and/or hyperparameter tuning
2. Direction-agnostic (standard dev of +3 just as anomalous as -3)
3. Alert if anomalous in only one dimension

DAS: simple, new method that overcomes these 3 problems
DAS: Directed Anomaly Scoring

1. Security analysts w/ limited time: specify $B = \text{alert budget}$

2. For set of events, assign each a “suspiciousness” score

3. Rank events by their “suspiciousness”

4. Output the $B$ most suspicious events for security team
DAS: Directed Anomaly Scoring

• Score(Event X) = # other events as benign as X in every dimension
  (Equivalent: # events where X at least as suspicious in every dim.)
DAS: Directed Anomaly Scoring

• Score(Event \( X \)) = \# other events as **benign** as \( X \) in every dimension
  (Equivalent: \# events where \( X \) at least as suspicious in every dim.)
DAS: Directed Anomaly Scoring

- Score(Event X) = # other events as **benign** as X in **every** dimension
  (Equivalent: # events where X **at least as suspicious** in every dim.)
DAS: Directed Anomaly Scoring

• Score(Event X) = # other events as **benign** as X in **every** dimension
  (Equivalent: # events where X **at least as suspicious** in every dim.)
Detection Results

• Real-time detector on 370 million emails over ~4 years

• Ran detector w/ total budget of 10 alerts/day
  • Practical for LBNL’s security team (~240 alerts/day typical)

• Detected 17 / 19 spearphishing attacks (89% TP)
  • 2 / 17 detected attacks were previously undiscovered
    (some were clicks where user didn’t enter creds)

• Best classical anom. det. : 4/19 attacks for same budget
  • Need budget >= 91 alerts/day to detect same # of attacks as DAS
Results: Cost of False Positives

• **10 alarms / day:** How much time does this cost the security team?

• Manually investigated all the alerts:
  • Avg. rate: 24 alerts / minute
  • **< 15 minutes** for analyst to investigate alerts for an entire month

• Subject / “From:” / URL = quick semantic filter
Results: Cost of False Positives

• **10 alarms / day:** How much time does this cost the security team?

• Manually investigated all the alerts:
  • Avg. rate: 24 alerts / minute
  • < **15 minutes** for analyst to investigate alerts for an entire month

• **Subject / “From:” / URL = quick semantic filter**
  • Never Lose Your Keys, Wallet, or Purse Again!
  • Invitation to Speak at Summit for Energy...
Results: Cost of False Positives

• **10 alarms / day:** How much time does this cost the security team?

• Manually investigated all the alerts:
  • Avg. rate: 24 alerts / minute
  • < 15 minutes for analyst to investigate alerts for an entire month

• Subject / “From:” / URL = *quick semantic filter*
  • Never Lose Your Keys, Wallet, or Purse Again!
  • Invitation to Speak at Summit for Energy...
  • From: Sales <Sales@gobblydegook.com>
Results: Cost of False Positives

• **10 alarms / day:** How much time does this cost the security team?

• Manually investigated all the alerts:
  • Avg. rate: 24 alerts / minute
  • <15 minutes for analyst to investigate alerts for an entire month

• **Subject / “From:” / URL = quick semantic filter**
  • Never Lose Your Keys, Wallet, or Purse Again!
  • Invitation to Speak at Summit for Energy...
  • From: Sales <Sales@gobblydegook.com>
  • http://lbl.XYZ.email-lists.com/ ...
  • http://secure-hotels.net/FooCompany/ListOfReservations.aspx?Lang=eng
Summary

• Real-time detecting of credential spearphishing attacks
  • TP = 89%: detects known + previously undiscovered attacks
  • FP = 0.004%: 10 alerts / day (< minutes per day)

• Bro implementation: contact asharma@lbl.gov

Key ideas

1. Leverage lure + exploit structure of spearphishing to design features

2. DAS: unsupervised, non-parametric technique for anomaly detection
   a) Generalizes beyond spearphishing
   b) “Needle-in-haystack” problems w/ curated & directional features
Real-time Detector: High Level Sketch

1. Get $B = \text{daily alert budget.}$

2. Extract lure + exploit features from emails as they arrive.

3. Do some tricks to setup DAS for real-time scoring.

4. When click-in-email event occurs, run real-time DAS for alert decision.