The Middler Reloaded

It's Not Just for the Web Anymore

Def Con 17

Jay Beale and Justin Searle

InGuardians
and
The Middler Project

The Middler – Application-Layer Rootkit

- Is a next-generation man-in-the-middle tool that takes the focus beyond the raw mechanics of the protocol.
- Targets the user's web applications with plug-ins that are specific to each app, abusing his privilege.
- Allows the attacker to modify the victim's interaction with the application bi-directionally, altering his reality.
- Example:
  - Make sure that he doesn't get to see that e-mail from his girlfriend
  - Send her e-mails from him, hiding them from his view of his Sent mail, and then deleting them before he sees them.

New Features

We're going to demo our new features today:

- Using a new protocol to put the victim in the Matrix: Voice over IP
- GUI for interactive victim selection and attack
- Generic session cloning tool for HTTP
- App-specific plug-ins
- Performance improvements
- Collaboration with Sophsec / libPoison

Adding Protocols

We initially targeted only applications that were accessed via HTTP.

We could do some neat tricks, like hooking the onkeypress() handler for an authentication form that loaded cleartext but will send the password via SSL.

The password gets sent to our server, one key at a time, while the user is still entering it!

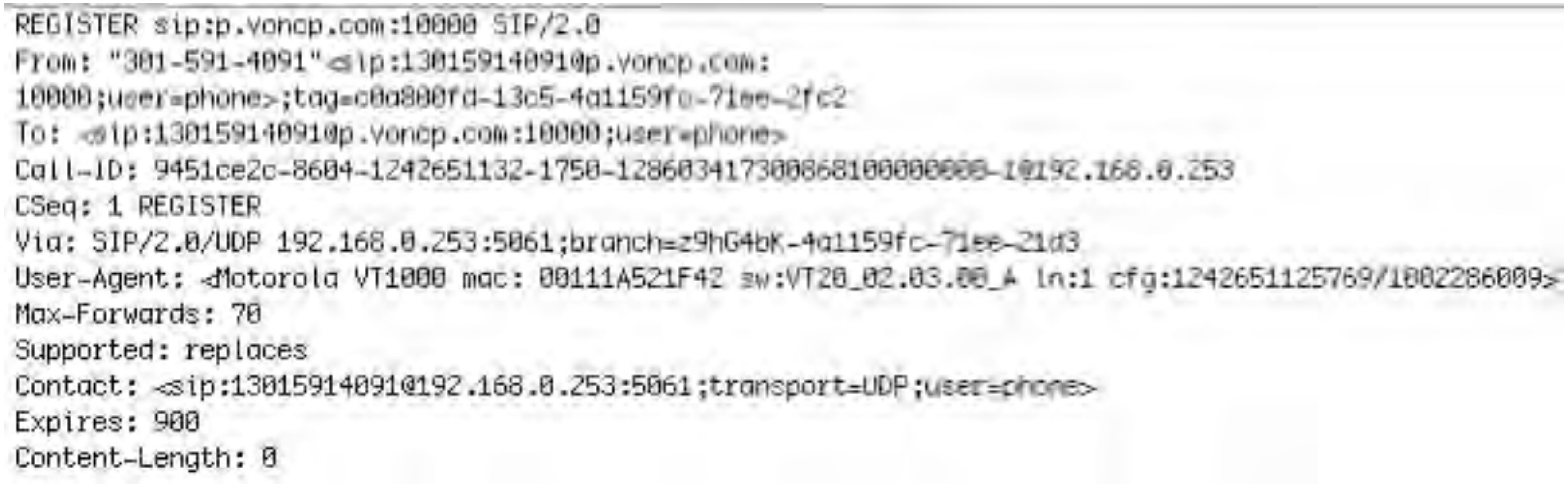
But everyone at Def Con kept telling us that we should start attacking non-web applications and protocols as well.

We decided to start with voice over IP (VoIP), for reasons that will become obvious in a moment.

Screwing Around with your Phone

Have you ever noticed how much SIP, the dominant Voice over IP protocol, looks like HTTP?

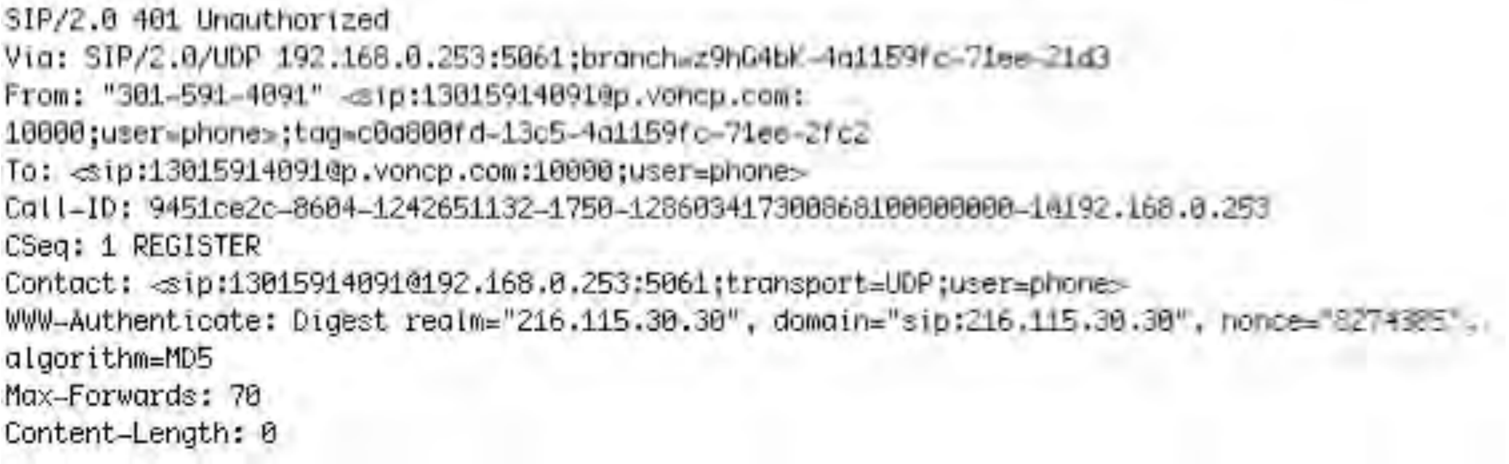
Here’s a request:



```
REGISTER sip:p.voncp.com:10000 SIP/2.0
From: "301-591-4091"<sip:13015914091@p.voncp.com:
10000;user=phone>;tag=c0a800fd-13c5-4a1159fc-71ee-2fc2
To: <sip:13015914091@p.voncp.com:10000;user=phone>
Call-ID: 9451ce2c-8604-1242651132-1750-1286034173008681000000000-1@192.168.0.253
CSeq: 1 REGISTER
Via: SIP/2.0/UDP 192.168.0.253:5061;branch=z9hG4bK-4a1159fc-71ee-21d3
User-Agent: <Motorola VT1000 mac: 00111A521F42 sw:VT20_02.03.00_A ln:1 cfg:1242651125769/1002286009>
Max-Forwards: 70
Supported: replaces
Contact: <sip:13015914091@192.168.0.253:5061;transport=UDP;user=phone>
Expires: 900
Content-Length: 0
```

Another Cleartext Protocol?

Here’s a response.



```
SIP/2.0 401 Unauthorized
Via: SIP/2.0/UDP 192.168.0.253:5061;branch=z9hG4bK-4a1159fc-71ee-21d3
From: "301-591-4091" <sip:13015914091@p.voncp.com:
10000;user=phone>;tag=c0a800fd-13c5-4a1159fc-71ee-2fc2
To: <sip:13015914091@p.voncp.com:10000;user=phone>
Call-ID: 9451ce2c-8604-1242651132-1750-12860341730086810000000-1@192.168.0.253
CSeq: 1 REGISTER
Contact: <sip:13015914091@192.168.0.253:5061;transport=UDP;user=phone>
WWW-Authenticate: Digest realm="216.115.30.30", domain="sip:216.115.30.30", nonce="8274385",
algorithm=MD5
Max-Forwards: 70
Content-Length: 0
```

Yes. We’ve found another helpless unencrypted protocol!

What about that Digest Authentication?

Yes, there's challenge-response authentication, with MD5 hashes.

While we could crack the password, the point is that we don't have to.

Even better than with HTTP cookies, most SIP implementations don't use a password after the session has been authenticated.

Hacking VoIP

Here are The Middler's current VoIP attacks.

- Send all of the inbound calls coming to a company or individual to your own phone instead.
- Redirect all of the inbound calls to your own device
- Alter incoming caller ID to confuse the victim
- Add the attacker's phone to the simul-ring list, so the attacker can listen in
- Alter sound of voice by mixing audio in
- Remove phone's registration so that it doesn't get calls.

Redirect Incoming Calls

The Middler can let you simply and easily redirect SOME but NOT ALL of the victim's incoming calls.

Interactively choose calls to redirect

OR

Set up a list of calls that the Middler should automatically redirect.

Alter Incoming Caller ID

The Middler can change the caller ID details without breaking the call.

Let’s confuse Jay. We’ll make his calls from his friend look like a telemarketer.

Interactively alter the caller ID

OR

Set up a list of numbers that the Middler should automatically rewrite.

Eavesdrop and Alter

Justin can eavesdrop on Jay’s call with Brandon

What if we could go a step further?

Justin might mix in audio telling Jay that the call is breaking up...

Brandon never hears that part – Justin could impersonates Jay

Brandon never knows the difference!

Demo: Unregister the Phone

Justin can make Jay’s calls just stop working.

He can unregister Jay’s phone and stop forwarding Jay’s phone’s registrations.

Let’s see this in action.

More New Features

Let’s look at the rest of those new features:

- GUI for interactive victim selection and attack
- Generic session cloning tool for HTTP
- App-specific plug-ins
- Performance improvement
- Collaboration with Sophsec via libPoison

New Feature: GUI Mode

The Middler now has a GUI, to help you choose our victim carefully.

Allows an attacker to find and carefully choose a target, based on the target's online identities.

- Webmail identity
- Social networking site identity

- Next: Cleartext POP/IMAP identity?

Once we select our target, what do we do with him?

Demo: GUI

Let’s demonstrate the GUI here.

We’ve set up a network here, with The Middler running.

Let’s see what users it has already identified.

Impersonating the User

The Middler lets you clone the user’s session, performing actions in parallel with the user, unbeknownst to him.

On social networking sites and e-mail sites, we can subtly do what only worms did before:

- Read his Inbox
- Delete messages from his InBox
- Add messages that were never truly sent
- Create a trust relationship, abuse it, and then remove it so he never knows it existed.
- Gather full contact information for his entire network

Web Applications with Plug-Ins

The Middler can let you do these kinds of attacks to these kind of applications:

- Social Networking
  - Twitter
  - Facebook
  - LinkedIn
- Web-based E-Mail Portals
  - Yahoo Mail
  - Gmail

The problem with all of these sites is that they take the user back to cleartext after authentication.

Web-based E-Mail Portals

Post-authentication, we can:

- Read the user’s e-mail
- Harvest the address book
- Send our own e-mails
- Profile the user in other applications
- Prevent a real logout, presenting the user with an actual logout but continuing the user’s session.

Think about how much information these portals have.

Social Networking Sites

On social networking sites, people have strong expectations that they can keep friend/network-only posts private, readable only by their friends or even small subsets of those friends.

As an attacker watching or middling a social networking session after it moves back into cleartext, you can:

- Read the user’s private entries
- Make the user’s private entries public
- Harvest the friends’ private entries
- Add your own user to the victim’s “friend” list

Cloning Arbitrary Sessions

The Middler also has features that aren’t application-specific.

The Middler is an HTML-aware proxy that you can set to act automatically whenever a session matches a pattern or whenever you interactively choose.

First of all, The Middler can clone any user’s session for you. This is normally a very manual process.

More Attacks on the Browser

The Middler can inject:

- Javascript
- IFRAMEs
- HTTP response code redirects
- HTML META redirects
- Any content you want, in replacement for any other content!

It can also gain subtle control of the browser itself, by forcing an IFRAME to surf to the Browser Exploitation Framework (BeEF)

For somewhat less subtlety, the Middler can make that IFRAME go to a Metasploit client-side exploit.

Why Does This All Work?

Many companies/developers don’t understand that leaving the post-login session unencrypted allows the attacker to impersonate the user.

For example, if you use your https://www.linkedin.com bookmark, click on “Sign In,” you’ll be taken to this URL:

```
https://www.linkedin.com/secure/login?trk=hb_signin
```

But then after login you’re taken to this cleartext one:

```
http://www.linkedin.com/home
```

Can’t I Change It Back?

You can change the URL to:

https://www.linkedin.com/home

…but clicking on any link will just take you right back to an HTTP URL!

Unless you modify your browser or surf with a special defensive proxy, you’ll constantly be pulling down cleartext links.

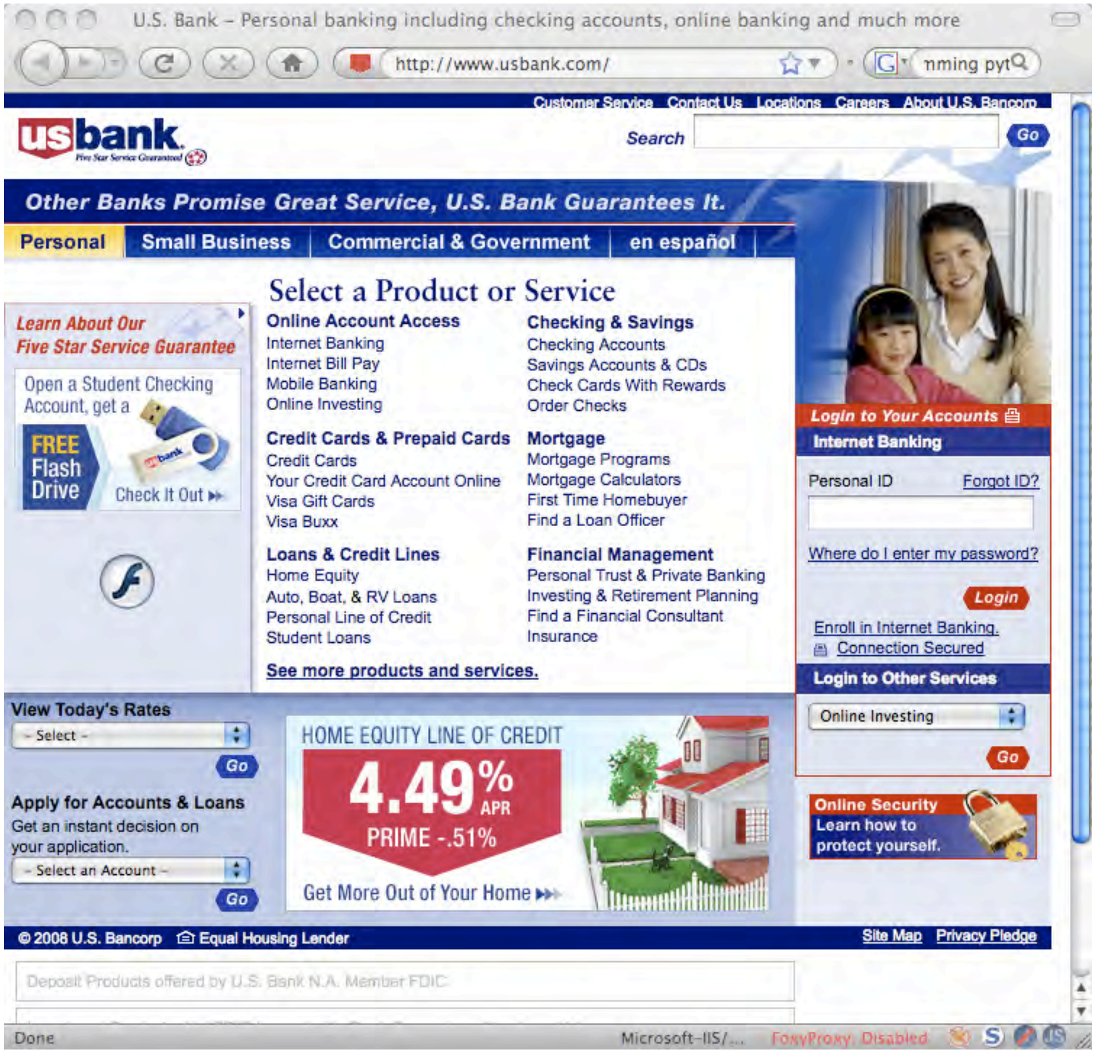
And it only takes one of those for me to start launching these kinds of attacks.

What about Banks?

There are even still banks that encrypt everything once you log in, but feed you the initial login form cleartext.

The problem for them is that if we can modify that initial page, then we can decide when to take the connection to SSL.

What’s Wrong With This Picture?



Other Banks Promise Great Service, U.S. Bank Guarantees It.

Learn About Our Five Star Service Guarantee

Open a Student Checking Account, get a FREE Flash Drive Check It Out ▸▸

Select a Product or Service

Online Account Access
Internet Banking
Internet Bill Pay
Mobile Banking
Online Investing

Checking & Savings
Checking Accounts
Savings Accounts & CDs
Check Cards With Rewards
Order Checks

Credit Cards & Prepaid Cards
Credit Cards
Your Credit Card Account Online
Visa Gift Cards
Visa Buxx

Mortgage
Mortgage Programs
Mortgage Calculators
First Time Homebuyer
Find a Loan Officer

Loans & Credit Lines
Home Equity
Auto, Boat, & RV Loans
Personal Line of Credit
Student Loans

Financial Management
Personal Trust & Private Banking
Investing & Retirement Planning
Find a Financial Consultant
Insurance

See more products and services.

Login to Your Accounts

Internet Banking

Personal ID Forgot ID?

Where do I enter my password?

Enroll in Internet Banking.
Connection Secured

Login to Other Services

Online Investing

View Today's Rates
- Select -

Apply for Accounts & Loans
Get an instant decision on your application.
- Select an Account -

HOME EQUITY LINE OF CREDIT
4.49% APR
PRIME -.51%
Get More Out of Your Home ▸▸

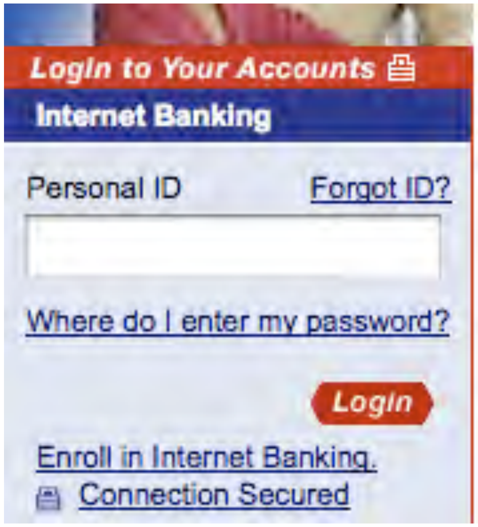
Online Security
Learn how to protect yourself.

© 2008 U.S. Bancorp Equal Housing Lender

Deposit Products offered by U.S. Bank N.A. Member FDIC

Cleartext Front Page

But it says our connection will be secured!



Login to Your Accounts

Internet Banking

Personal ID

Forgot ID?

Where do I enter my password?

Login

Enroll in Internet Banking.

Connection Secured

<form … action=https://www4.usbank.com/internetBanking/LoginRouter

What if I were to re-write all the URLs on this site to remove the SSL?

CSRF's Attacks Made Easier?

Imagine a non-security friend on a hotel network...

He types the name of his online banking site into his browser:

http://www.usbank.com

He's used to the bank protecting him from himself. The site reloads the page with an HTTPS version:

https://www.usbank.com.

It's already too late. It's a race condition and he lost.

Demo: Hooking onKeyPress()

I have already served him my own front page for the HTTP site, modified on the fly from the bank’s real page, but with links changed and/or JavaScript inserted.

We have a cool way of taking advantage of this. What if we hook the keypress events, the same way other AJAX applications do to do word completion?

Let’s demonstrate!

Knowing Where the User Has Been

We get powers from middling a user that you normally associate with Cross Site Scripting vulnerabilities.

For example, I can read the browser history to see what links the user has visited.

If the victim has pop-blocking in place, I can even just inject Javascript into any HTTP-carried pages the user has open.

Developing Plugins

We write The Middler in Python, using a plug-in architecture.

It’s pretty easy to develop plug-ins. Come join us!

Credits

The Middler Project has gained more developers:

- Tom Liston, exploiter of virtual machines, creator of the JavaScript password keystroke logger
- Matt Carpenter, Python god and speeder of code
- Brandon Edwards and Sophsec, creators of libPoison, the low-level network capture code that's getting us our own hostile DHCP server.
- Tyler Reguly, beta-testing, soon to be working on software installation and update with Brandon

Speaker Bio: Justin Searle

Justin Searle, a Senior Security Analyst with InGuardians, specializes in penetration testing and security architecture. Previously, Justin served as JetBlue Airway's IT Security Architect and has provided top-tier support for the largest supercomputers in the world. Justin has taught hacking techniques, forensics, networking, and intrusion detection courses for multiple universities and corporations. Justin has presented at top security conferences including DEFCON, ToorCon, ShmooCon, and SANS. In his rapidly dwindling spare time, Justin co-leads prominent open source projects including The Middler, Samarai Web Testing Framework, and the social networking pentest tools: Yokoso! and Laudnum. He is actively working to finish the upcoming bestseller the Seven Most Deadly Social Network Hacks, with Tom Eston of the Security Justice Podcast, and Kevin Johnson of InGuardians. Justin has an MBA in International Technology and is GIAC-certified in incident handling and hacker techniques (GCIH) and intrusion analysis (GCIA).

Speaker Bio: Jay Beale

Jay Beale is an information security specialist, well known for his work on threat avoidance and mitigation technology. He's written two of the most popular security hardening tools used worldwide throughout industry and government: Bastille UNIX, a system lockdown and audit tool that introduced a vital security-training component, and the Center for Internet Security's Unix Scoring Tool. Through Bastille and his work with the Center, Jay has provided leadership in the space of proactive system security. Jay also contributed to the OVAL project and the Honeynet Project.Jay has served as an invited speaker at a variety of conferences and government symposia worldwide. He's written for Information Security Magazine, SecurityFocus, and SecurityPortal. Jay has co-authored or edited nine books in the Information Security space, including six in his Open Source Security Series and two technical works of fiction in the "Stealing the Network" series. Jay is a senior security analyst and managing partner at InGuardians, where he gets to work with brilliant people on topics ranging from application penetration to virtual machine escape. Prior to this, Jay served as the Security Team Director for MandrakeSoft, helping set company strategy, design security products, and pushing security into the then third largest retail Linux distribution.