ENPM 809W Introduction to Secure Software Engineering

Gananand Kini

Lecture 7

Authentication and Authorization related security bugs - Attacks

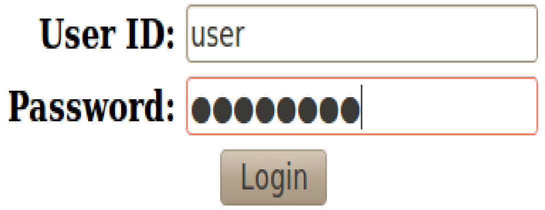
Outline

- **Authentication core concepts**
  - Something you know, something you have and something you are
  - Password based authentication
    - Weak password requirements
    - Weak password recovery mechanism
- **Authentication Weaknesses**
- **Authorization core concepts**
- **Authorization Weaknesses**
  - Confused Deputy Problem
  - Third party services authorization
- **Time of check Time of Use problems in Authentication & Authorization**

Authentication core concepts

Authentication Core Concepts

- **Authentication: Verifying the *identity* of a user, process, or device, often as a pre-requisite to allowing access to resources in an information system.**
- **Two factor authentication: Leverage at least two of the methods below for a single authentication transaction.**



Something you know

Something you have

Something you are

Sources:
1. Fingerprint Scanner Image: Licensed under Creative Commons Attribution – Share Alike 3.0 Unported– Created by Rachmaninoff. Retrieved September 21, 2011, from https://secure.wikimedia.org/wikipedia/commons/wiki/File:Fingerprint_scanner_identification.jpg
2. NIST Glossary: Authentication. https://csrc.nist.gov/glossary/term/authentication. Retrieved July 10, 2021.

Bypassing authentication

- **More about verifying identity and the actor or entity is who/what they say they are.**
- **The following can lead to authentication bypass vulnerabilities:**
  - Poor authentication mechanism and implementation
  - Weak passwords
  - Weak identity mechanisms
  - Poor password policies and storage practices.
  - Many others…
  - But again … passwords…

Sources:
1. Albert Gareev. The checkmarks are not what we should be looking for. http://automation-beyond.com/2016/06/02/the-checkmarks-are-not-what-we-should-be-looking-for/. Retrieved July 10, 2021.

Authentication Weaknesses

More issues with authentication

- After authentication, how long is the user going to remain authenticated?
- Can another user masquerade using current or another user/entity's identity?
- Are there information leaks that can help bypass authentication?

Sources:
1. Neon Light Doormat Home Décor Speak Friend and Enter. https://www.batenstore.com/products/speak-friend-and-enter-doormat-front-door-mat-best-door-mat-practical-housewarming-gift-idea?variant=1000006297023627. Retrieved October 10, 2021.

CWE-307: Improper Restriction of Excessive Authentication Attempts

- **Description: The software does not implement sufficient measures to prevent multiple failed authentication attempts within in a short time frame, making it more susceptible to brute force attacks.**

- **You: “Aha! You tried to sneak past without speaking ‘friend’ to enter…”**

- **Attacker: “Oh well.. I’ll just try that again...”**

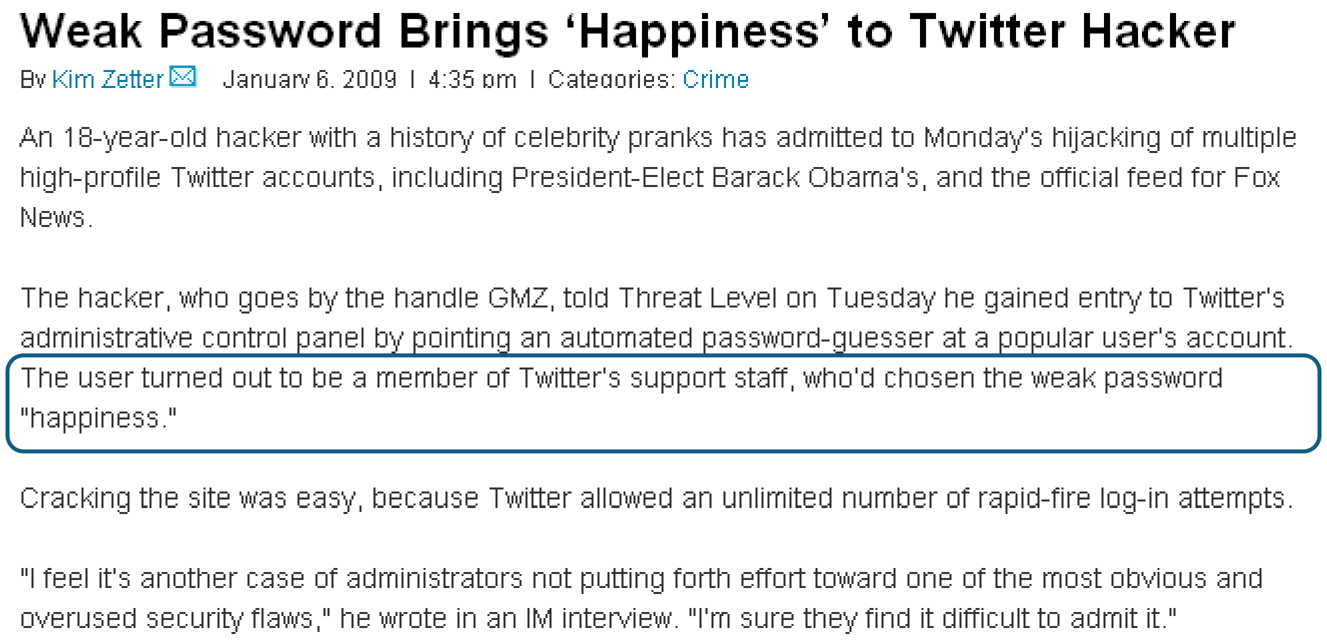
Sources:
1. CWE-307: Improper Restriction of Excessive Authentication Attempts. MITRE CWE. https://cwe.mitre.org/data/definitions/307.html. Retrieved July 20, 2021.

Excessive Logins Example: No restrictions

```c
int validateUser (char *host, int port)
{
  int isValidUser = 0;

  char username[USERNAME_SIZE];
  char password[PASSWORD_SIZE];

  while (isValidUser == 0)
  {
    if (getUserInput(username,USERNAME_SIZE) == 0) error();
    if (getUserInput(password,PASSWORD_SIZE) == 0) error();

    isValidUser = AuthenticateUser (username, password);
  }

  return(SUCCESS);
}
```

The validateUser() method will continuously check for a valid username and password without any restriction on the number of authentication attempts made.

Real World Example - Twitter



Weak Password Brings 'Happiness' to Twitter Hacker

By Kim Zetter   January 6, 2009 | 4:35 pm | Categories: Crime

An 18-year-old hacker with a history of celebrity pranks has admitted to Monday's hijacking of multiple high-profile Twitter accounts, including President-Elect Barack Obama's, and the official feed for Fox News.

The hacker, who goes by the handle GMZ, told Threat Level on Tuesday he gained entry to Twitter's administrative control panel by pointing an automated password-guesser at a popular user's account. The user turned out to be a member of Twitter's support staff, who'd chosen the weak password "happiness."

Cracking the site was easy, because Twitter allowed an unlimited number of rapid-fire log-in attempts.

"I feel it's another case of administrators not putting forth effort toward one of the most obvious and overused security flaws," he wrote in an IM interview. "I'm sure they find it difficult to admit it."

Zetter, K. (2009, January 6). *Weak password brings 'happiness' to Twitter hacker.* Retrieved February 3, 2011, from http://www.wired.com/threatlevel/2009/01/professed-twitt/

CWE-521: Weak Password Requirements

- Description: The product does not require that users should have strong passwords, which makes it easier for attackers to compromise user accounts.
- Authentication mechanisms often rely on a memorized secret (also known as a password) to provide an assertion of identity for a user of a system. It is therefore important that this password be of sufficient complexity and impractical for an adversary to guess.
- The specific requirements around how complex a password needs to be depends on the type of system being protected. Selecting the correct password requirements and enforcing them through implementation are critical to the overall success of the authentication mechanism.

Sources:
1. CWE-521: Weak Password Requirements. MITRE CWE. https://cwe.mitre.org/data/definitions/521.html. Retrieved July 20, 2021.

Real World Example – Xbox One “Secure” Login (2014)

- Kristoffer von Hassel (5 years old) entered wrong password for his dad’s account on login screen.
- This took him to an alternate verification screen.
- Obtained access by filling alternate verification password field with lots of spaces.
- Acknowledged in Microsoft’s Security acknowledgements!

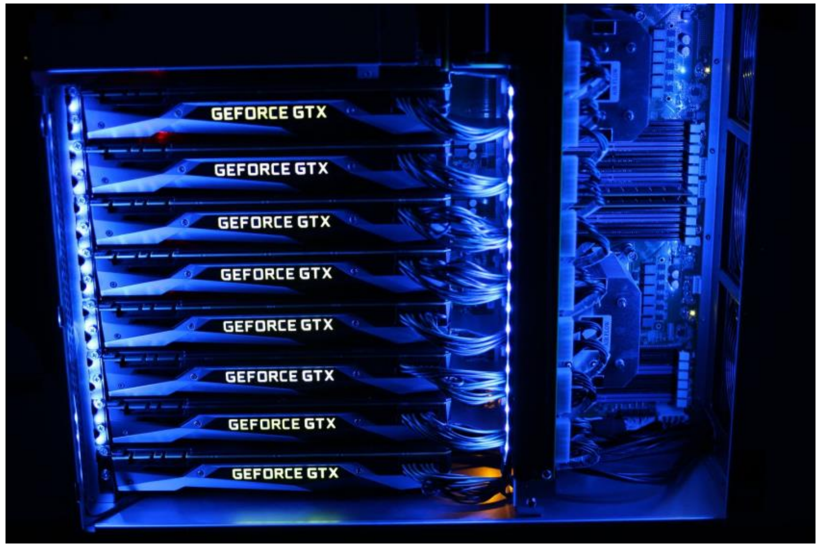
Sources:
1. Kid bypasses Xbox password security with one simple trick. https://www.slashgear.com/kid-bypasses-xbox-password-security-with-one-simple-trick-04323809/. Retrieved July 10, 2021.

Poor choice of hashing, salt and password policies

- **MD5 and SHA-1 are broken. Can find collisions easily.**
- **The birthday attack makes it easier to reduce the search space for collisions.**
- **Remember the HBGary CMS story?**
  - The passwords were using MD5 hashes and the rainbow tables at the time were not up to par.
  - Still would have given time for them to recover.
  - Unfortunately, the real issue was that the passwords used were too simple!

Password cracking today

- **A lot of utilities available to break weak encryption, password hash schemes like:**
  - hashcat
  - John the ripper
  - Etc.
- **These utilities are already GPU accelerated.**
- **Can help inform and craft password policies for choice of minimum password strength for your application or software system design.**
- **Source (images on right): https://www.shellntel.com/blog/2019/2/19/how-to-build-a-2nd-8-gpu-password-cracker**



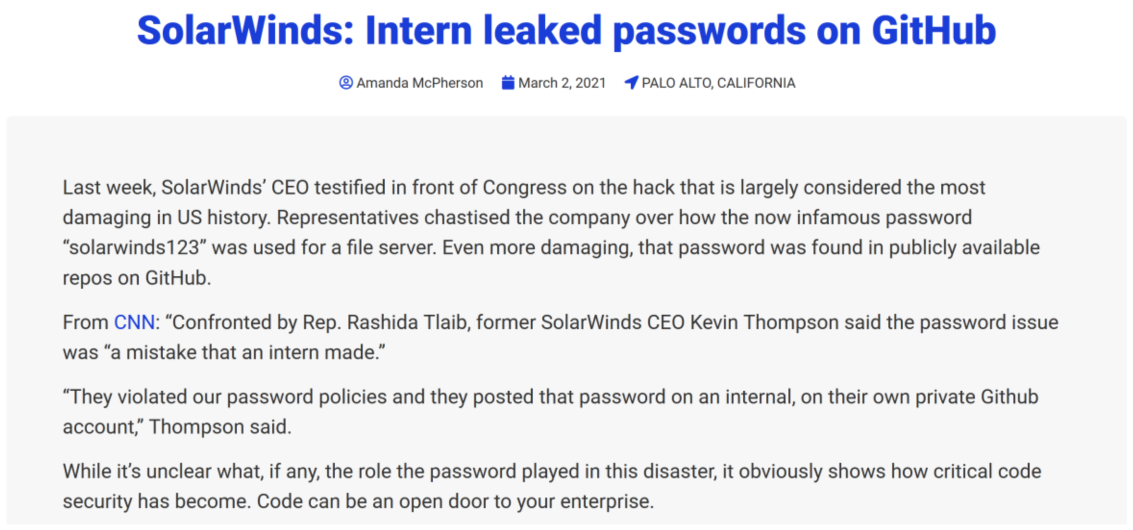
CWE-259: Use of Hard-coded Password

- **Description: The software contains a hard-coded password, which it uses for its own inbound authentication or for outbound communication to external components.**
- **A hard-coded password typically leads to a significant authentication failure that can be difficult for the system administrator to detect. Once detected, it can be difficult to fix, so the administrator may be forced into disabling the product entirely. There are two main variations:**
  - Inbound: the software contains an authentication mechanism that checks for a hard-coded password.
  - Outbound: the software connects to another system or component, and it contains hard-coded password for connecting to that component.

Sources:
1. CWE-259: Use of Hard-coded Password. MITRE CWE. https://cwe.mitre.org/data/definitions/259.html. Retrieved July 20, 2021.

Other authentication issues

- **Hardcoded storage of passwords in the source code or source control repository.**
  - This is really bad! See [1].
- **Craig Hays actually tried to follow up after password was leaked accidentally [2].**
  - Instead of just resetting credentials, he setup another cloud platform based SSH honeypot (Cowrie) with leaked credentials to see what would happen.
  - Found 2 clones of the dummy repository with leaked credentials.
  - Had at least 18 views of the password file.
  - 2 people tried to install a botnet malware.
  - 1 person tried to exfiltrate sensitive information.
  - 1 person had a look around and then deleted history to cover their tracks.
  - 2 people just tried logging in and looking around.



SolarWinds: Intern leaked passwords on GitHub

Amanda McPherson March 2, 2021 PALO ALTO, CALIFORNIA

Last week, SolarWinds' CEO testified in front of Congress on the hack that is largely considered the most damaging in US history. Representatives chastised the company over how the now infamous password "solarwinds123" was used for a file server. Even more damaging, that password was found in publicly available repos on GitHub.

From CNN: "Confronted by Rep. Rashida Tlaib, former SolarWinds CEO Kevin Thompson said the password issue was "a mistake that an intern made."

"They violated our password policies and they posted that password on an internal, on their own private Github account," Thompson said.

While it's unclear what, if any, the role the password played in this disaster, it obviously shows how critical code security has become. Code can be an open door to your enterprise.

Sources:
1. Amanda McPherson. SolarWinds: Intern leaked passwords on Github. https://blubracket.com/solarwinds-intern-leaked-passwords-on-github/. March 2, 2021. Retrieved July 10, 2021.
2. Craig Hays. What happened when I Leaked My Server Password on Github.com .https://craighays.com/what-happened-when-leaked-server-password-github/. June 10, 2020. Retrieved July 10, 2021.

Other authentication issues

- Many software systems implement password recovery mechanisms.

- Even if you implement a strong password scheme and login, a poorly designed or implemented password recovery scheme can have disastrous security consequences!

CWE-640: Weak Password Recovery Mechanism for Forgotten Password

- **Description: The software contains a mechanism for users to recover or change their passwords without knowing the original password, but the mechanism is weak.**
- **It is common for an application to have a mechanism that provides a means for a user to gain access to their account in the event they forget their password.**
- **Very often the password recovery mechanism is weak, which has the effect of making it more likely that it would be possible for a person other than the legitimate system user to gain access to that user's account.**
- **Weak password recovery schemes completely undermine a strong password authentication scheme.**

Sources:
1. CWE-640: Weak Password Recovery Mechanism for Forgotten Password. MITRE CWE. https://cwe.mitre.org/data/definitions/640.html. Retrieved July 20, 2021.

Real World Example - eBay

A famous example of this type of weakness being exploited is the eBay attack. eBay always displays the user id of the highest bidder. In the final minutes of the auction, one of the bidders could try to log in as the highest bidder three times. After three incorrect log in attempts, eBay password throttling would kick in and lock out the highest bidder's account for some time. An attacker could then make their own bid and their victim would not have a chance to place the counter bid because they would be locked out. Thus an attacker could win the auction.

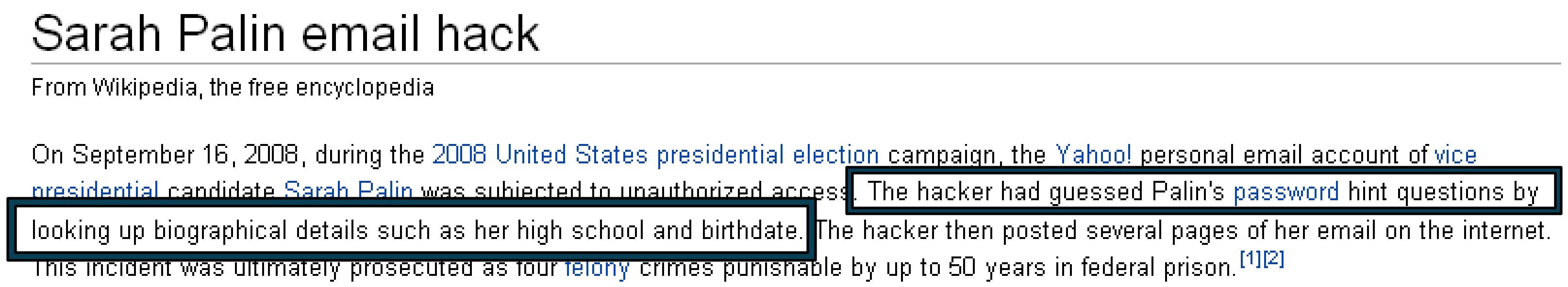
Mitigations:

- ➢ Shorten the length of account lockout
- ➢ Don't show who the highest bidder is
- ➢ Don't expose user id, only expose name
    - ○ Name should never be used as a key

Real World Example – Yahoo! & Sarah Palin

Yahoo! email used three security questions:

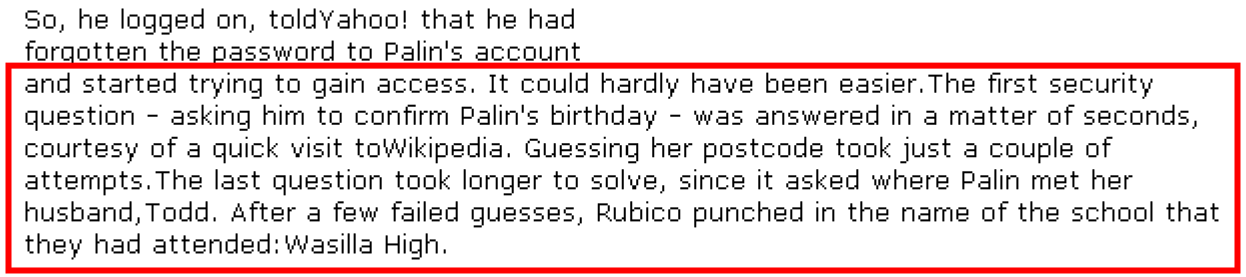
1. Birthday
2. Zip code
3. Where she met her husband



Sarah Palin email hack

From Wikipedia, the free encyclopedia

On September 16, 2008, during the 2008 United States presidential election campaign, the Yahoo! personal email account of vice presidential candidate Sarah Palin was subjected to unauthorized access. The hacker had guessed Palin's password hint questions by looking up biographical details such as her high school and birthdate. The hacker then posted several pages of her email on the internet. This incident was ultimately prosecuted as four felony crimes punishable by up to 50 years in federal prison.[1][2]

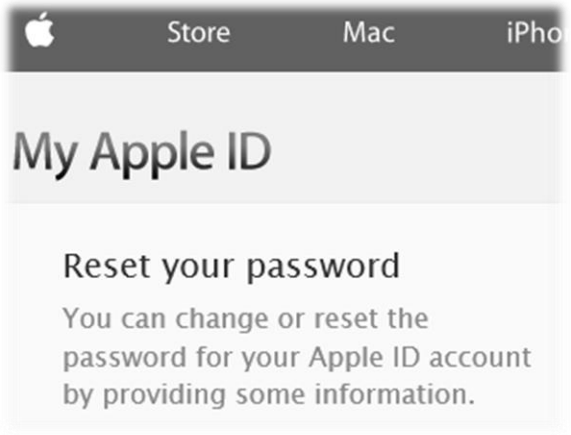


So, he logged on, toldYahoo! that he had forgotten the password to Palin's account and started trying to gain access. It could hardly have been easier.The first security question - asking him to confirm Palin's birthday - was answered in a matter of seconds, courtesy of a quick visit toWikipedia. Guessing her postcode took just a couple of attempts.The last question took longer to solve, since it asked where Palin met her husband,Todd. After a few failed guesses, Rubico punched in the name of the school that they had attended:Wasilla High.

Sarah Palin email hack. (2010, May 26). Retrieved June 2, 2010, from http://en.wikipedia.org/wiki/Sarah_Palin_email_hack
Johnson, B. (2010, May 23). *Sarah.palin@hacked-off.com*. Retrieved June 3, 2010, from http://findarticles.com/p/news-articles/sunday-telegraph-the-london-uk/mi_8064/is_20100523/sarahpalinhacked-offcom/ai_n53726137/

Real World Example – Apple iForgot

1) iforgot.apple.com – enter Apple ID
2) Select authentication method – “answer security questions”
3) Enter date of birth
4) Answer two security questions
5) Enter new password
6) Password is reset



My Apple ID

Reset your password

You can change or reset the password for your Apple ID account by providing some information.

Knowing someone’s Apple ID and DOB would allow construction of the URL after step #5.

The exploit was published on the day that Apple launched two-factor authentication for Apple ID accounts, which would have prevented the attack for anyone that had enabled it. Once activated, the feature replaces the security question based verification with a 4-digit code sent to the user's mobile device

Welch, C. (2013, March 13). *Major security hole allows Apple passwords to be reset with only email address, date of birth.* Retrieved November 5, 2014, from http://www.theverge.com/2013/3/22/4136242/major-security-hole-allows-apple-id-passwords-reset-with-email-date-of-birth

Authorization core concepts

What is authorization?

- **Authorization: The right, permission, or privilege granted to system entity to access a system resource or functionality.**
- **Asks the following questions:**
  - What level of access to software system resources or functionality is necessary for the user or entity?
  - If it is role based, are they granular enough to support only the necessary functionality?
- **Treat the software system holistically – are all parts or functionality covered?**

Authorization Weaknesses

Authorization weaknesses

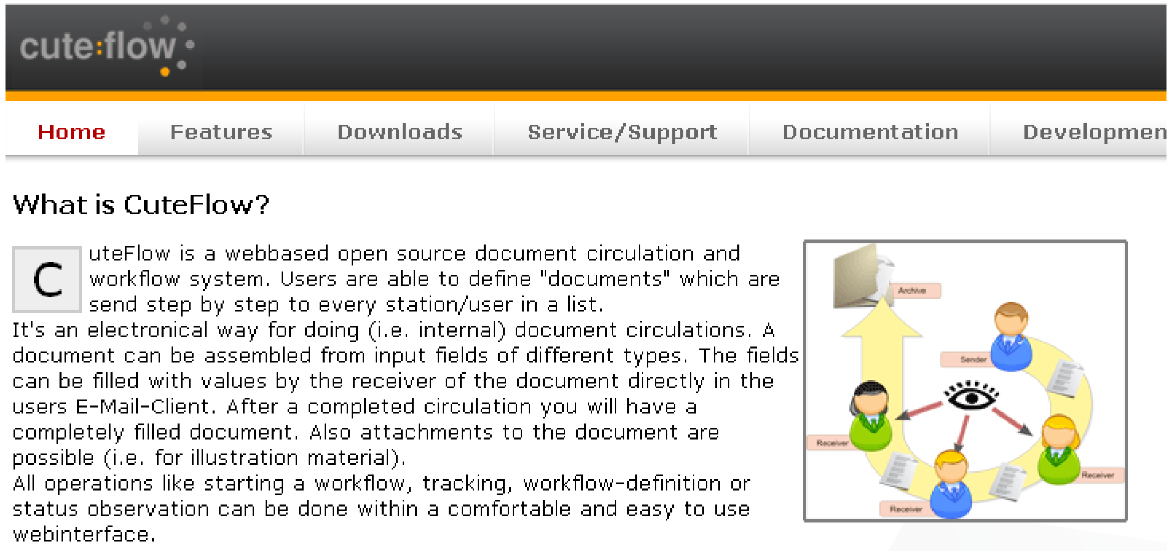
- **Authorization bypasses can occur in many different ways:**
  - Authorization checks are either absent or not implemented correctly
  - Not all resources are protected with authorization checks
  - A confused deputy is used to provide authorization to resources that otherwise was unavailable to an attacker

CWE-425: Direct Request ('Forced Browsing')

- Description: The web application does not adequately enforce appropriate authorization on all restricted URLs, scripts, or files.
- Web applications susceptible to direct request attacks often make the false assumption that such resources can only be reached through a given navigation path and so only apply authorization at certain points in the path.
- Don't use the term 'forced browsing' since it can have multiple contexts.
- Essentially the software system allows an unauthorized user to access resources (like pages) when directly browsed to it by a user because it is missing an authorization check or the routing for authorization has not been implemented correctly.

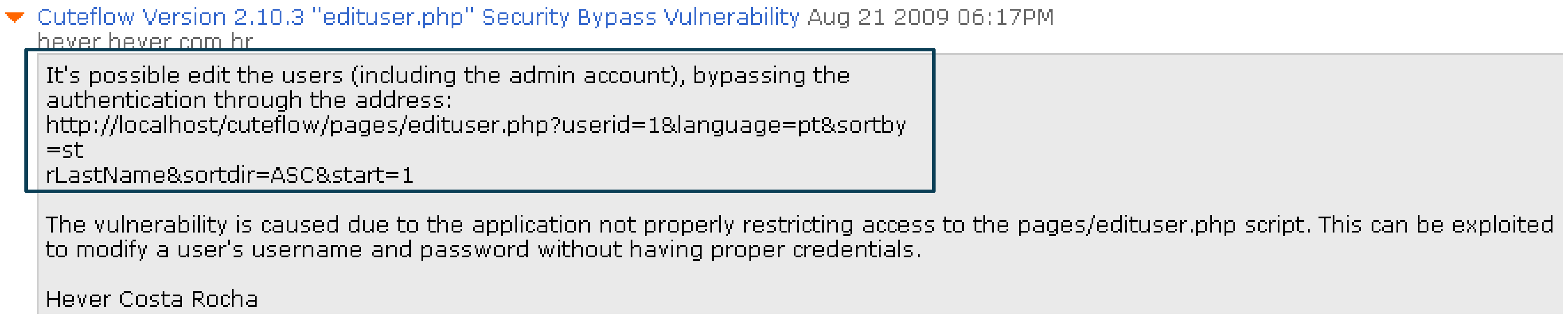
Sources:
1. CWE-425: Direct Request ('Forced Browsing'). MITRE CWE. https://cwe.mitre.org/data/definitions/435.html. Retrieved July 20, 2021.

Real World Example – CuteFlow Exploit



What is CuteFlow?

CuteFlow is a webbased open source document circulation and workflow system. Users are able to define "documents" which are send step by step to every station/user in a list.
It's an electronical way for doing (i.e. internal) document circulations. A document can be assembled from input fields of different types. The fields can be filled with values by the receiver of the document directly in the users E-Mail-Client. After a completed circulation you will have a completely filled document. Also attachments to the document are possible (i.e. for illustration material).
All operations like starting a workflow, tracking, workflow-definition or status observation can be done within a comfortable and easy to use webinterface.



Cuteflow Version 2.10.3 "edituser.php" Security Bypass Vulnerability Aug 21 2009 06:17PM
hever hever com br

It's possible edit the users (including the admin account), bypassing the authentication through the address:
http://localhost/cuteflow/pages/edituser.php?userid=1&language=pt&sortby
=st
rLastName&sortdir=ASC&start=1

The vulnerability is caused due to the application not properly restricting access to the pages/edituser.php script. This can be exploited to modify a user's username and password without having proper credentials.

Hever Costa Rocha

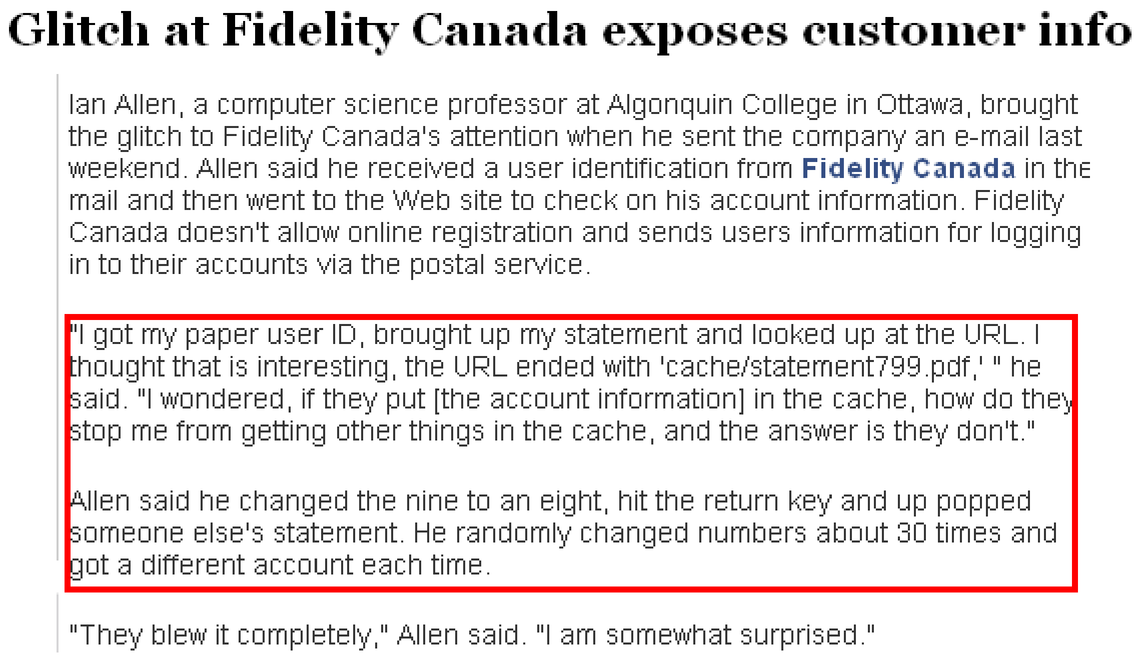
Rocha, Hever. (2009, August 21). Cuteflow version 2.10.3 "edituser.php" security bypass vulnerability. Retrieved February 11, 2011, from http://www.securityfocus.com/archive/1/506000/100/0/threaded

CWE-639: Authorization Bypass Through User-Controlled Key

- **Description: The system's authorization functionality does not prevent one user from gaining access to another user's data or record by modifying the key value identifying the data.**
- **Assuming:**
  - Resources are retrieved using ‘keys’ (that implicitly act as intermediary authorization tokens)
- **Because/And:**
  - The authorization process does not properly check the entitlement or privileges of accessing that resource for a specific user/role or entity.
- **Then**
  - This allows for (potentially not necessarily) another user to access another user’s resource on the software system just by using the ‘key’.

Sources:
1. CWE-639: Authorization Bypass Through User-Controlled Key. MITRE CWE. https://cwe.mitre.org/data/definitions/639.html. Retrieved July 20, 2021.

Real World Example – Fidelity Canada



Glitch at Fidelity Canada exposes customer info

Ian Allen, a computer science professor at Algonquin College in Ottawa, brought the glitch to Fidelity Canada's attention when he sent the company an e-mail last weekend. Allen said he received a user identification from **Fidelity Canada** in the mail and then went to the Web site to check on his account information. Fidelity Canada doesn't allow online registration and sends users information for logging in to their accounts via the postal service.

"I got my paper user ID, brought up my statement and looked up at the URL. I thought that is interesting, the URL ended with 'cache/statement799.pdf,' " he said. "I wondered, if they put [the account information] in the cache, how do they stop me from getting other things in the cache, and the answer is they don't."

Allen said he changed the nine to an eight, hit the return key and up popped someone else's statement. He randomly changed numbers about 30 times and got a different account each time.

"They blew it completely," Allen said. "I am somewhat surprised."

Usually, when users can directly access a PDF or other non-code file from the web server, (e.g., resource is located in the web root) there is no opportunity for authorization code to execute.

With a predictable structure to the filename, it only takes minutes to create a script capable of retrieving all of the statements/reports on the site!

Sullivan, B. (2002, May 30). *Glitch at Fidelity Canada exposes customer info*. Retrieved June 3, 2010, from http://www.itworldcanada.com/news/glitch-at-fidelity-canada-exposes-customer-info/124086

Authorization failure example: Authorization leak

```
my $q = new CGI;

my $message_id = $q->param('id');

if (!AuthorizeRequest(GetCurrentUser())
{
   ExitError("not authorized to perform this function");
}

my $Message = LookupMessageObject($message_id);

print "From: " . encodeHTML($Message->{from}) . "<br>\n";
print "Subject: " . encodeHTML($Message->{subject});
print "\n<hr>\n";
print "Body: " . encodeHTML($Message->{body}) . "\n";
```

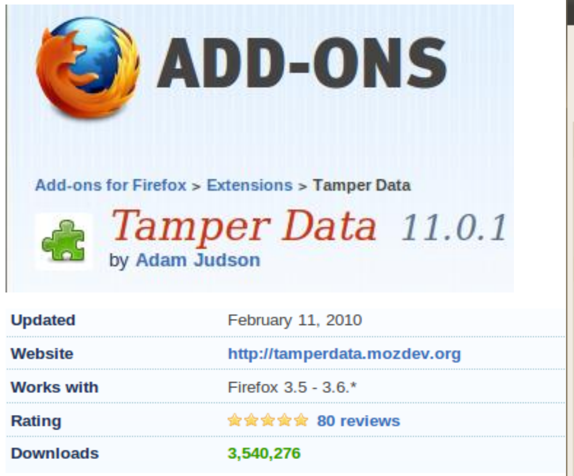
While the program properly exits if authentication fails, it does not ensure that the message is addressed to the user. As a result, an authenticated attacker could provide any arbitrary identifier and read private messages that were intended for other users.

CWE-602: Client-Side Enforcement of Server-Side Security

- **Description: The software is composed of a server that relies on the client to implement a mechanism that is intended to protect the server.**
- **When the server relies on protection mechanisms placed on the client side, an attacker can modify the client-side behavior to bypass the protection mechanisms resulting in potentially unexpected interactions between the client and server.**
- **The consequences will vary, depending on what the mechanisms are trying to protect.**

Sources:
1. CWE-602: Client-Side Enforcement of Server-Side Security. MITRE CWE. https://cwe.mitre.org/data/definitions/602.html. Retrieved July 20, 2021.

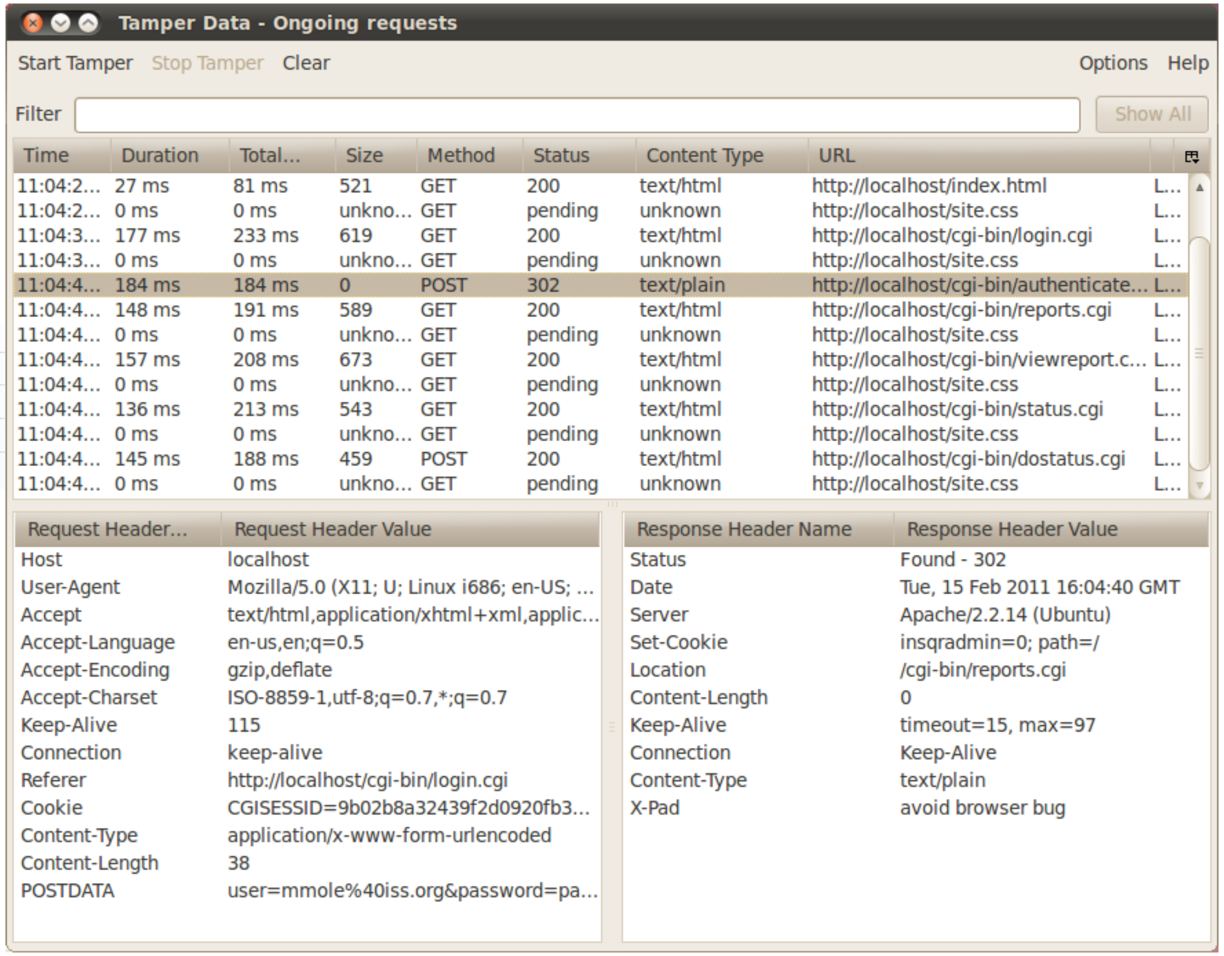
Client Enforced Security – Exploit Example



Add-ons for Firefox > Extensions > Tamper Data

Tamper Data 11.0.1
by Adam Judson

Updated	February 11, 2010
Website	http://tamperdata.mozdev.org
Works with	Firefox 3.5 - 3.6.*
Rating	★★★★★ 80 reviews
Downloads	3,540,276



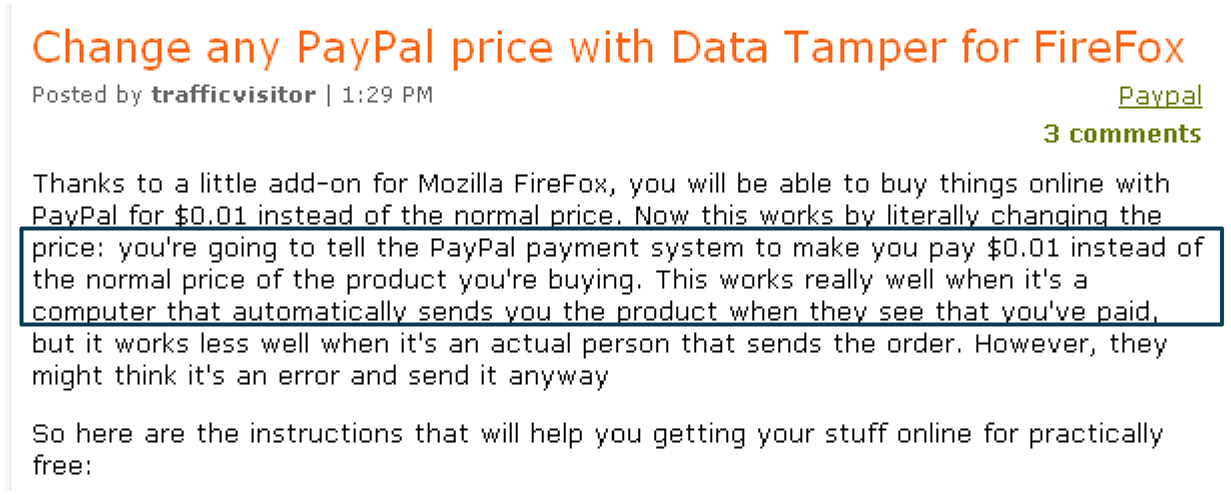
Time	Duration	Total...	Size	Method	Status	Content Type	URL	
11:04:2...	27 ms	81 ms	521	GET	200	text/html	http://localhost/index.html	L...
11:04:2...	0 ms	0 ms	unkno...	GET	pending	unknown	http://localhost/site.css	L...
11:04:3...	177 ms	233 ms	619	GET	200	text/html	http://localhost/cgi-bin/login.cgi	L...
11:04:3...	0 ms	0 ms	unkno...	GET	pending	unknown	http://localhost/site.css	L...
11:04:4...	184 ms	184 ms	0	POST	302	text/plain	http://localhost/cgi-bin/authenticate...	L...
11:04:4...	148 ms	191 ms	589	GET	200	text/html	http://localhost/cgi-bin/reports.cgi	L...
11:04:4...	0 ms	0 ms	unkno...	GET	pending	unknown	http://localhost/site.css	L...
11:04:4...	157 ms	208 ms	673	GET	200	text/html	http://localhost/cgi-bin/viewreport.c...	L...
11:04:4...	0 ms	0 ms	unkno...	GET	pending	unknown	http://localhost/site.css	L...
11:04:4...	136 ms	213 ms	543	GET	200	text/html	http://localhost/cgi-bin/status.cgi	L...
11:04:4...	0 ms	0 ms	unkno...	GET	pending	unknown	http://localhost/site.css	L...
11:04:4...	145 ms	188 ms	459	POST	200	text/html	http://localhost/cgi-bin/dostatus.cgi	L...
11:04:4...	0 ms	0 ms	unkno...	GET	pending	unknown	http://localhost/site.css	L...

Request Header...	Request Header Value
Host	localhost
User-Agent	Mozilla/5.0 (X11; U; Linux i686; en-US; ...
Accept	text/html,application/xhtml+xml,applic...
Accept-Language	en-us,en;q=0.5
Accept-Encoding	gzip,deflate
Accept-Charset	ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive	115
Connection	keep-alive
Referer	http://localhost/cgi-bin/login.cgi
Cookie	CGISESSID=9b02b8a32439f2d0920fb3...
Content-Type	application/x-www-form-urlencoded
Content-Length	38
POSTDATA	user=mmole%40iss.org&password=pa...

Response Header Name	Response Header Value
Status	Found - 302
Date	Tue, 15 Feb 2011 16:04:40 GMT
Server	Apache/2.2.14 (Ubuntu)
Set-Cookie	insqradmin=0; path=/
Location	/cgi-bin/reports.cgi
Content-Length	0
Keep-Alive	timeout=15, max=97
Connection	Keep-Alive
Content-Type	text/plain
X-Pad	avoid browser bug

Debug Proxies:
A key reason why a client cannot be entrusted with performing security.

Real World Example – PayPal & Vendor Issue


Change any PayPal price with Data Tamper for FireFox

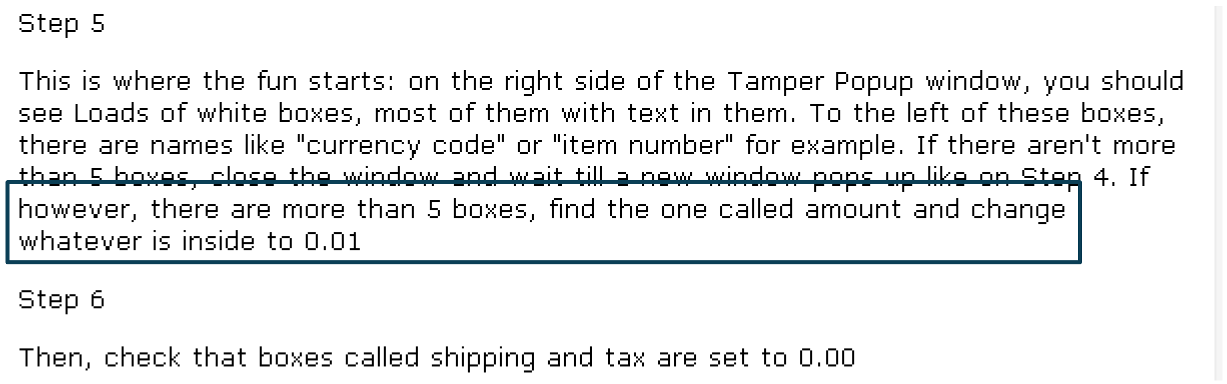
Posted by **trafficvisitor** | 1:29 PM

Paypal

3 comments

Thanks to a little add-on for Mozilla FireFox, you will be able to buy things online with PayPal for $0.01 instead of the normal price. Now this works by literally changing the price: you're going to tell the PayPal payment system to make you pay $0.01 instead of the normal price of the product you're buying. This works really well when it's a computer that automatically sends you the product when they see that you've paid, but it works less well when it's an actual person that sends the order. However, they might think it's an error and send it anyway

So here are the instructions that will help you getting your stuff online for practically free:


Step 5

This is where the fun starts: on the right side of the Tamper Popup window, you should see Loads of white boxes, most of them with text in them. To the left of these boxes, there are names like "currency code" or "item number" for example. If there aren't more than 5 boxes, close the window and wait till a new window pops up like on Step 4. If however, there are more than 5 boxes, find the one called amount and change whatever is inside to 0.01

Step 6

Then, check that boxes called shipping and tax are set to 0.00

Trafficvisitor. (2009, July). *Change any PayPal price with Data Tamper for FireFox*. Retrieved February 15, 2011, from http://letsearndollar.blogspot.com/2009/07/change-any-paypal-price-with-data.html

Authorization failure example: Client side authorization

Client

```
$customerid = AskForCustomerId();
$address = AskForAddress();

writeSocket($sock, "$user AUTH $customerid\n");
$resp = readSocket($sock);

if ($resp eq "authorized")
{
   # request is authorized, go ahead and update the info!
   writeSocket($sock, "$user CHANGE-ADDRESS $customerid $address\n");
}
else {  print "ERROR: You're not authorized to change customer's address.\n"; }
```

Server

```
($user, $cmd, $customerid, $address) = ParseRequest($sock);

if ($cmd eq "AUTH")
{
   $result = AuthorizeRequest($user, $customerid);
   writeSocket($sock, "$result\n");
}

elsif ($cmd eq "CHANGE-ADDRESS")
{
  if (validateAddress($address)) {
     $res = UpdateAddress($customerid, $address);
     writeSocket($sock, "SUCCESS\n");
  }
  else { writeSocket($sock, "FAILURE -- address is malformed\n"); }
}
```

An attacker can bypass authentication by just sending a CHANGE-ADDRESS command.

Authorization with third-party services

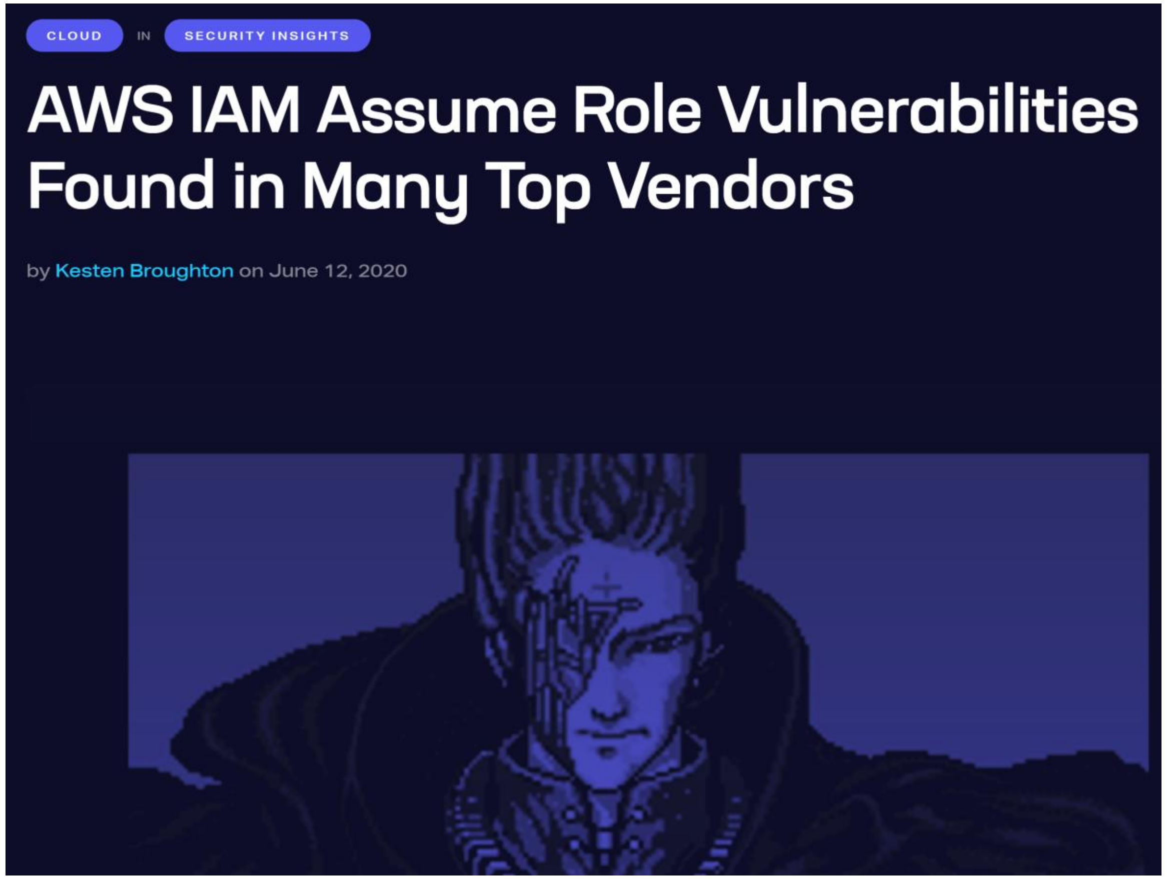
- **If your software system depends on the use of third-party services:**
  - What identity is being used to authenticate with the third-party service?
  - A malicious user could end up *misusing* this identity through your software system.
  - Which aspects of this third-party service is implicitly being authorized by your software system?

CWE-441: Unintended Proxy or Intermediary (‘Confused Deputy’)

- **Description: The product receives a request, message, or directive from an upstream component, but the product does not sufficiently preserve the original source of the request before forwarding the request to an external actor that is outside of the product's control sphere.**
- **This causes the product to appear to be the source of the request, leading it to act as a proxy or other intermediary between the upstream component and the external actor.**

Amazon AWS cloud `ExternalId` confused deputy problem

- **Amazon created `ExternalId` to mitigate confused deputy problems in its cloud services.**
- **Vendors that didn't understand misconfigured it and ended up causing confused deputy problems.**
- **External ID is supposed to be unique identifier that acts as a authority token for a third party to assume a role within service provider/vendor's AWS environment.**
- **Becomes even more of a problem with the "root" role (not super user with high privilege but rather a trusted account role that can delegate authority)**



AWS IAM Assume Role Vulnerabilities Found in Many Top Vendors

by Kesten Broughton on June 12, 2020

Sources:
1. Kesten Broughton. AWS IAM Assume Role Vulnerabilities Found in Many Top Vendors. https://www.praetorian.com/blog/aws-iam-assume-role-vulnerabilities/. June 12, 2020. Retrieved July 10, 2021.

Linux Kernel "Confused Deputy" fix

- Linux patched an issue (merged in version 5.13 on May 25, 2021) where a process could tell/"trick" another process to change its own process file's security attributes in /proc/$pid/attr/ via execve.
- See https://www.kernel.org/doc/html/latest/security/credentials.html?highlight=confused#open-file-credentials for more details.
- The fix now only allows changes when the file opener is the task itself and not another task.

Sources:

1. Cook, Kees. Proc: Check /proc/$pid/attr/ writes against file opener. https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=bfb819ea20ce8bbeeba17e1a6418bf8bda91fc28. Retrieved October 10, 2021.

2. Kernel.org. https://www.kernel.org/doc/html/latest/security/credentials.html?highlight=confused#open-file-credentials. Retrieved October 10, 2021.

Next time …

- **Authentication and Authorization Related Bugs – Defenses.**