



Security Analysis of Mattermost

By Changze Cui & Weihao Dong

What is Mattermost



What is Mattermost

The screenshot displays the Mattermost web interface. On the left is a blue sidebar with a list of channels and direct messages. A red rectangle highlights this sidebar. The main area shows the 'Town Square' channel feed, also outlined with a red rectangle. The feed includes a post from 'vegetable_chicken' with a long URL, a system message about a user being removed, and two tweets. The bottom of the interface features a text input field for sending messages to the channel.

Sidebar (Left):

- 261test @weihao
- PUBLIC CHANNELS +
 - Demo Plugin
 - Off-Topic
 - Town Square
 - More...
- PRIVATE CHANNELS +
 - Chess: Surveybot VS
- DIRECT MESSAGES +
 - github
 - surveybot
 - vegetable_chicken
 - More...
- Switch Channels - ⓂK

Channel: Town Square

vegetable_chicken 5:21 PM
https://www.facebook.com/plugins/post.php?href=https%3A%2F%2Fwww.facebook.com%2Fpermalink.php%3Fstory_fbid%3D551104579155306%26id%3D100027673065589&width=500

Sat, Apr 25, 2020

System 12:08 PM
@vegetable_chicken was removed from the team.

weihao 1:55 PM
<https://twitter.com/PebblesPuss2014/status/1251476543538331648>

Twitter
Princess Pebbles on Twitter
"Have a safe #Caturday everyone 🐾"

<https://twitter.com/joejothemainbro/status/1253463810272186368>

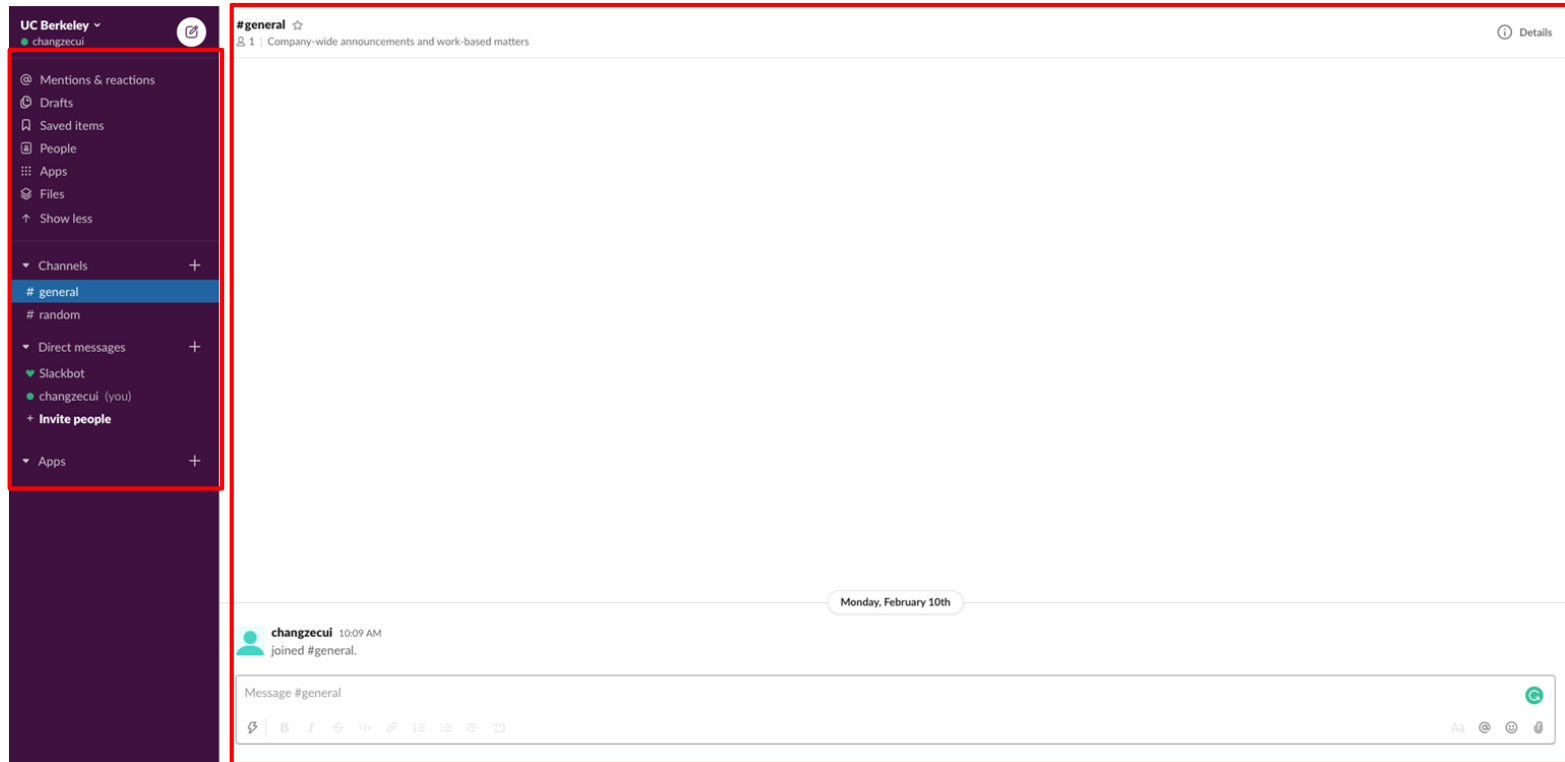
Twitter
Joseph on Twitter
"I have no words. This Travis Scott fortnite concert was spectacular. #fortnite"

Today


weihao 1:47 PM
<script>alert("hacked!!");

Write to Town Square

What is Mattermost























Key Difference between Mattermost and Slack

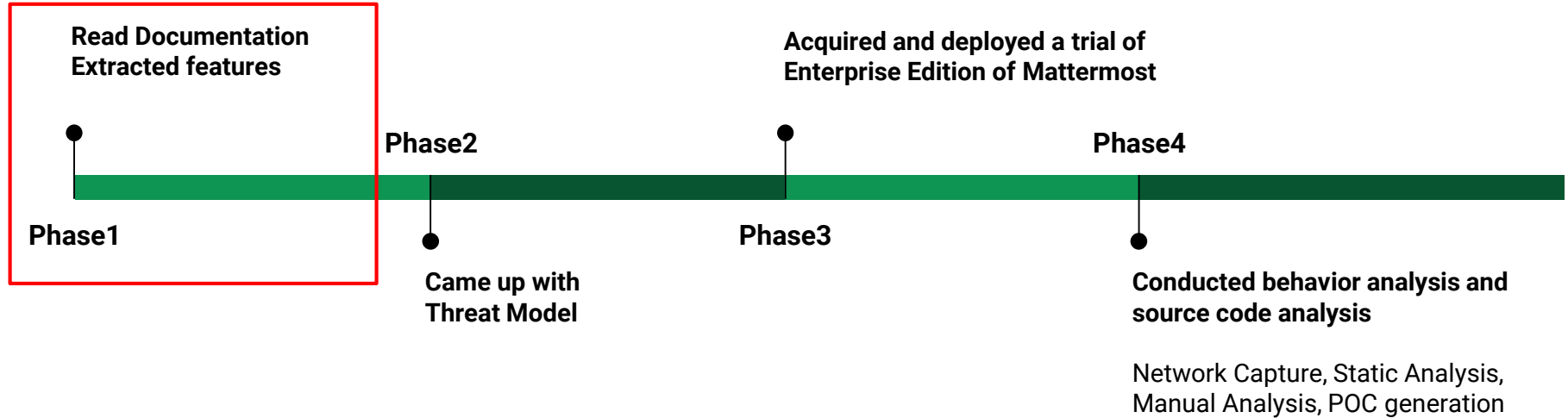
| | 
Mattermost |  |
|----------------------|--|---|
| Support Self-Hosting | yes | no |
| Open Source | yes | no |

Mattermost envisions itself as an open source Slack alternative

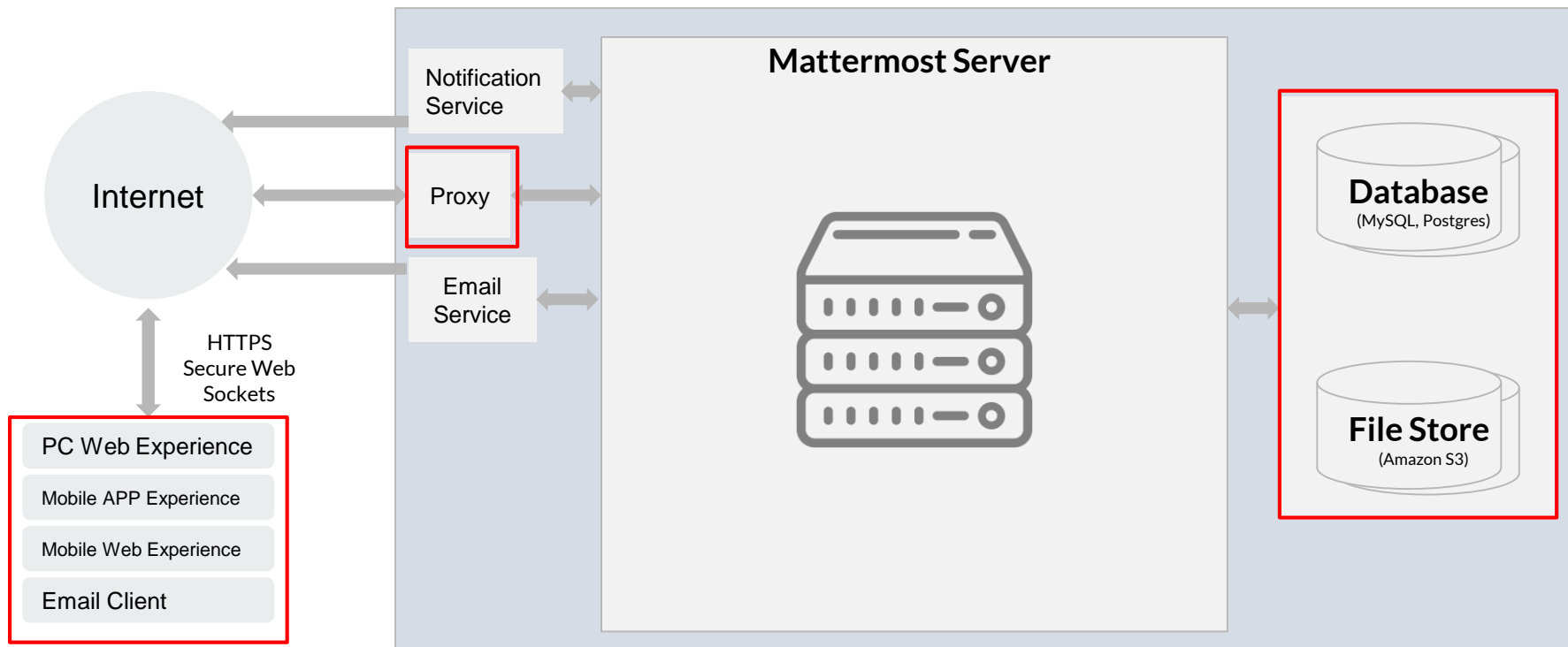
Motivations

| | | | | |
|---|---|--|---|--|
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

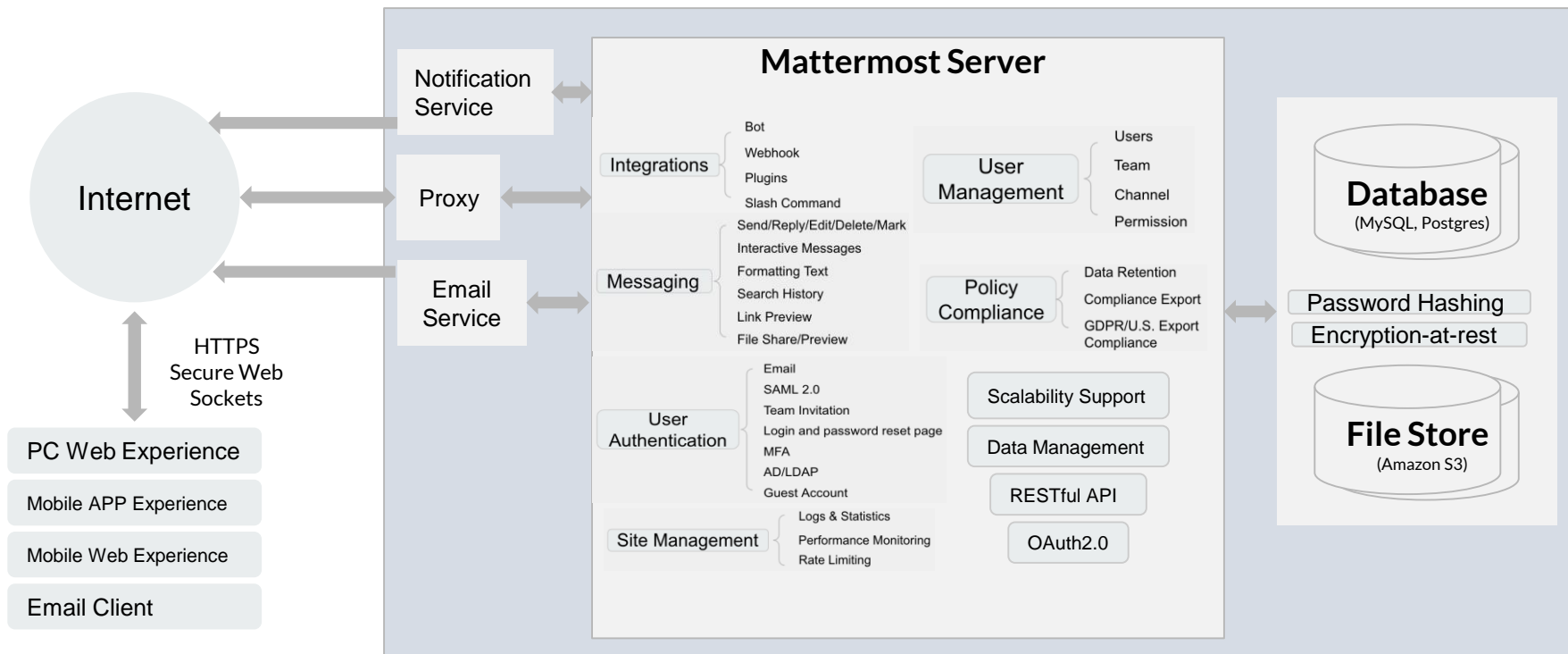
Project Timeline



Architecture of Mattermost

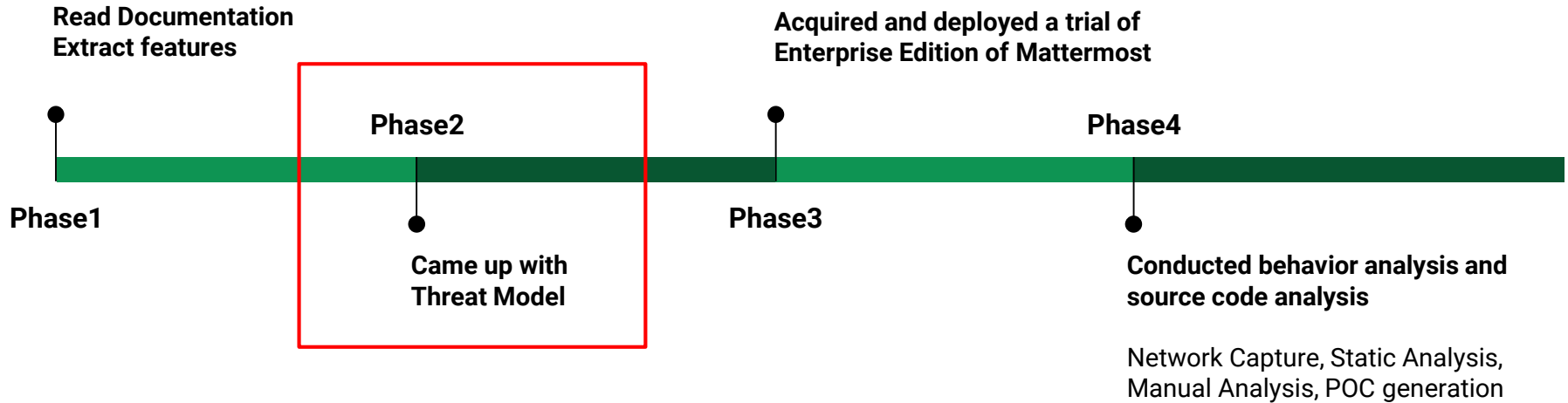


Features of Mattermost

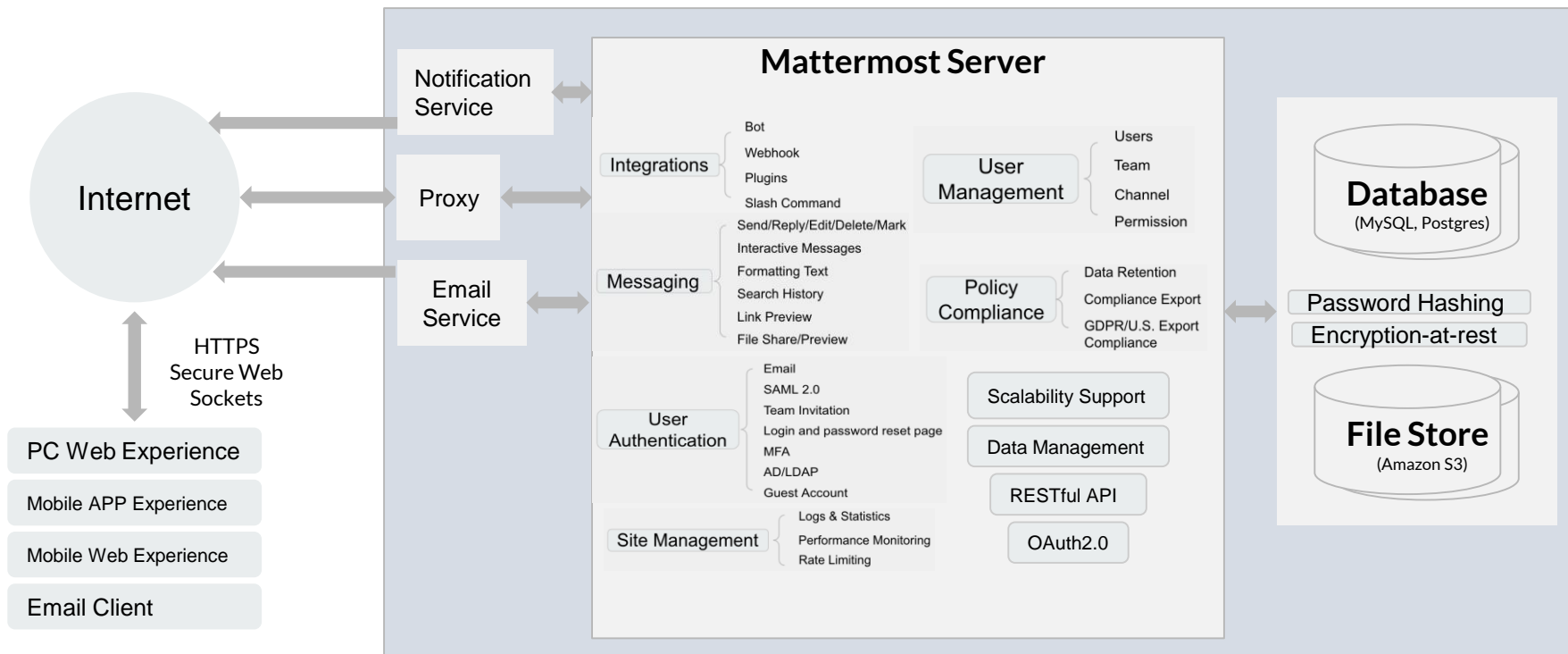




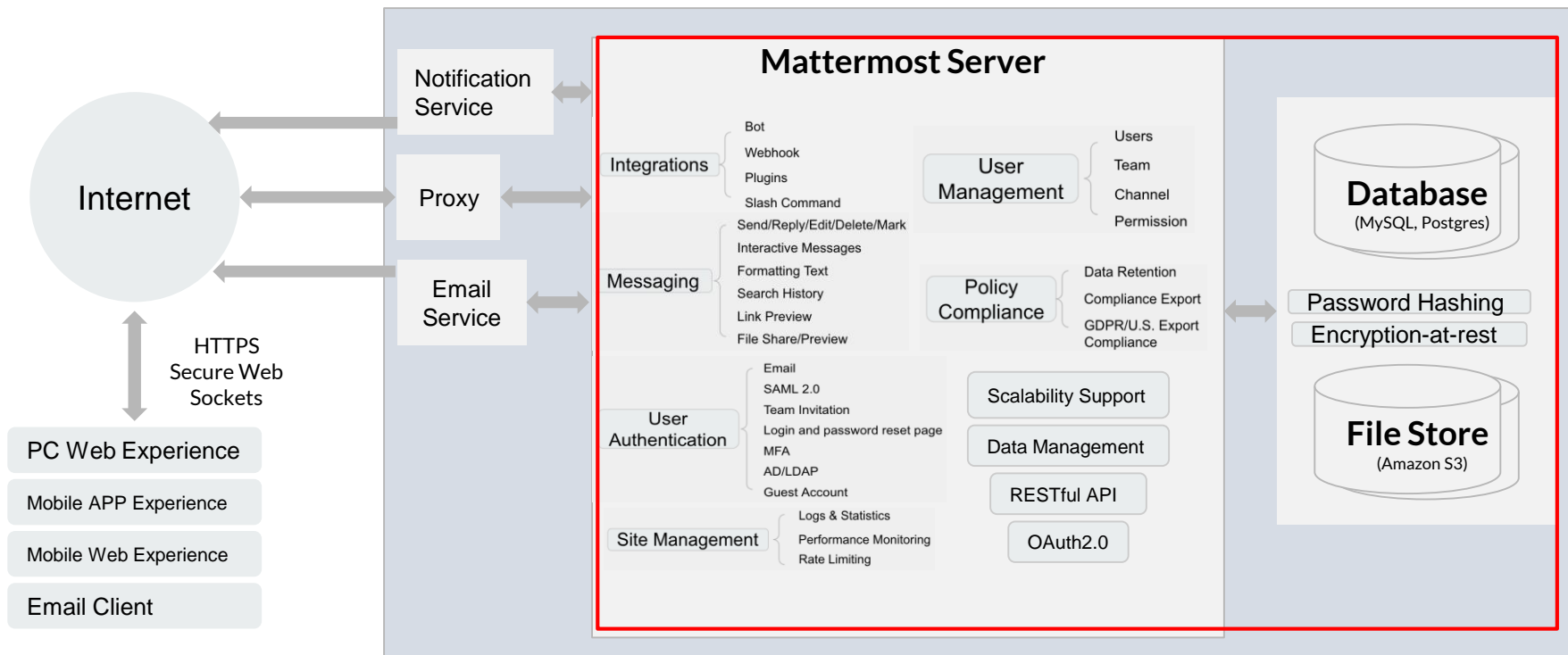
Project Timeline



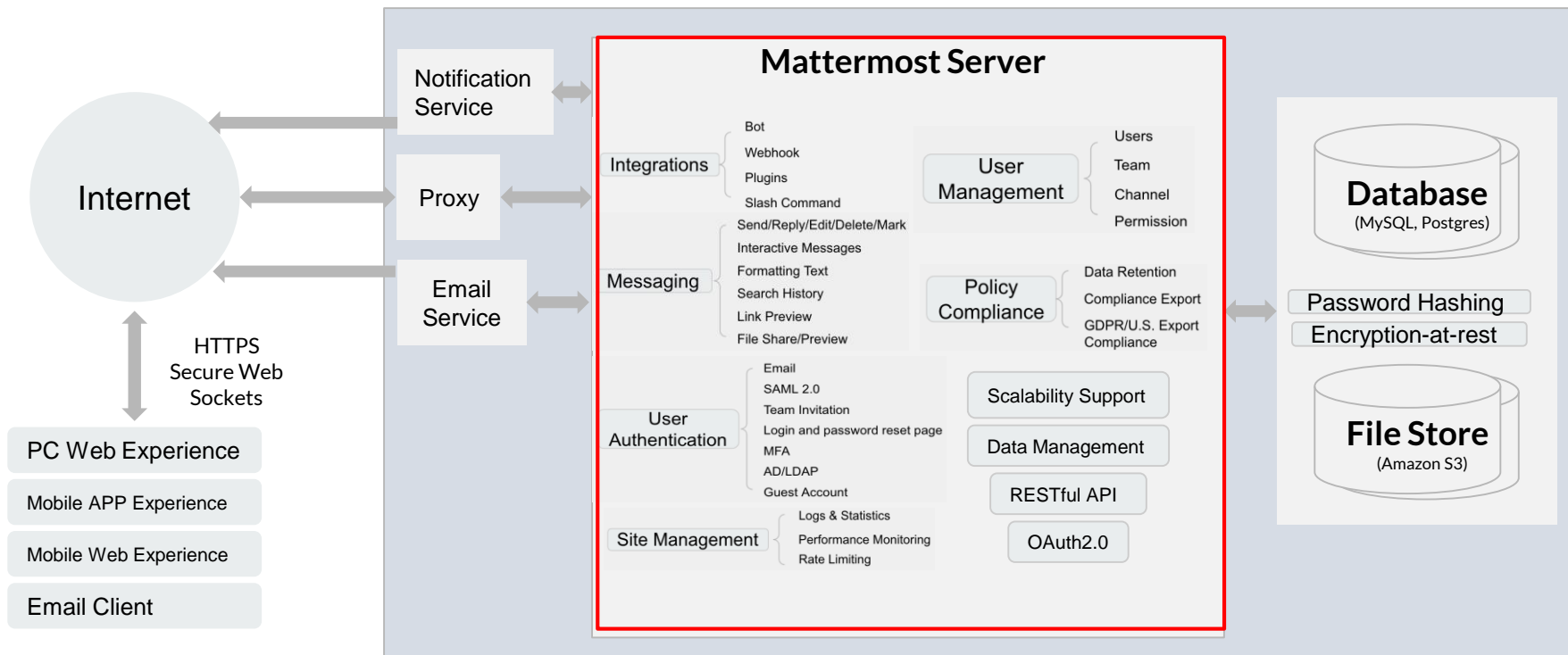
How We Come Up With Our Threat Model



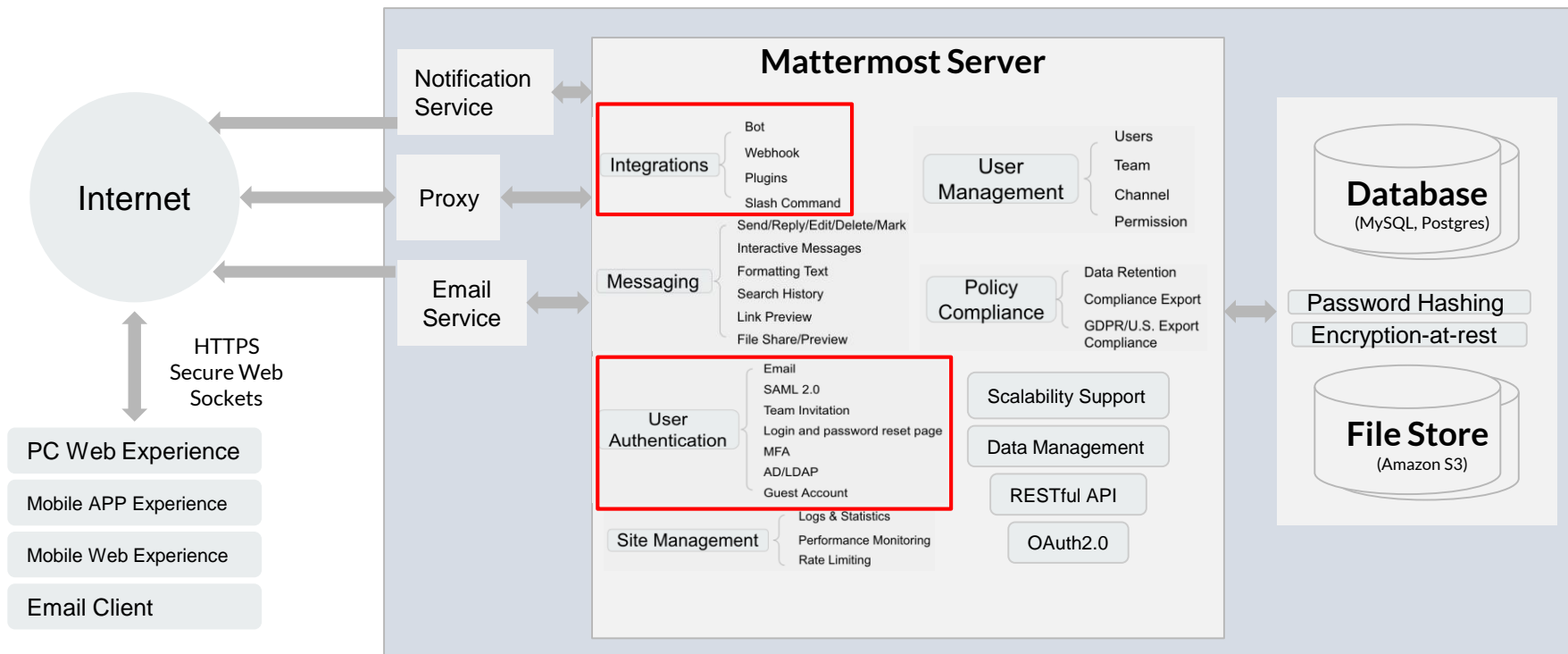
Attackers in Server & Database



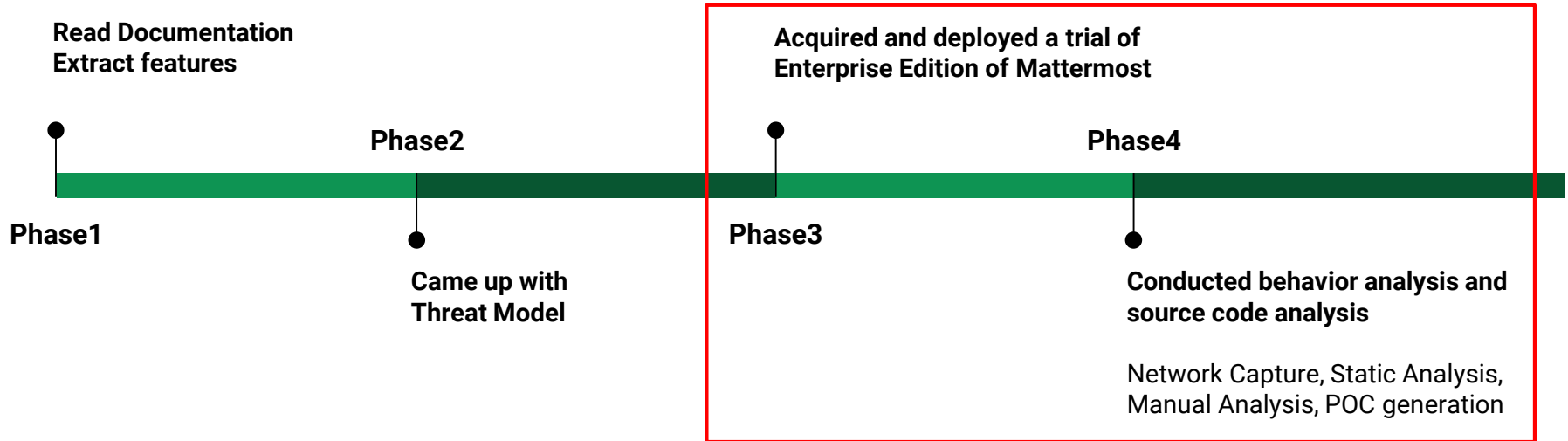
Attackers as Non-Admin Mattermost Users



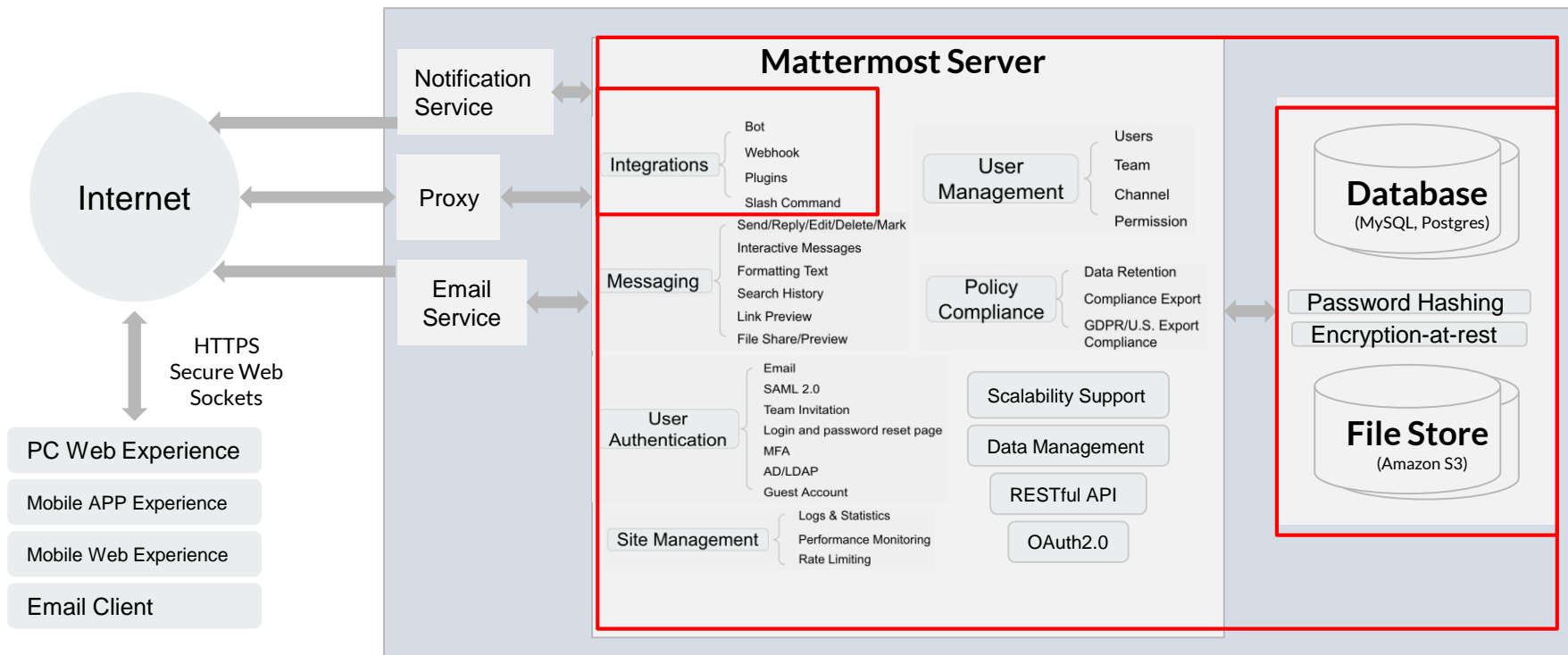
Attackers as Non-Users



Project Timeline



Attackers in Server & Database





Attackers in Server & Database

Everything is readable. But is it the end of the nightmare?

1. Credential breach
2. Session token stealing
3. Integrated applications can be compromised



Attackers in Server & DB - Credential Breach

Password

- Plain text passwords can be used for credential stuffing
- Mattermost only stores **bcrypt** hashed passwords into database
- bcrypt is probably not the best choice with the evolution of parallel hardware ¹
- scrypt and argon2 ² provide better defense against offline parallel cracking

[1] Malvoni, Katja, and Josip Knezovic. "Are your passwords safe: Energy-efficient bcrypt cracking with low-cost parallel hardware." 8th USENIX Workshop on Offensive Technologies (WOOT 14). 2014.

[2] Biryukov, Alex, Daniel Dinu, and Dmitry Khovratovich. "Argon2: new generation of memory-hard functions for password hashing and other applications." *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2016.

Attackers in Server & DB - Credential Breach

Password

- Plain text passwords can be used for credential stuffing

password-hashing.net

Password Hashing Competition
and our recommendation for hashing passwords: Argon2

[1] Malvoni, Katja, and Josip Knezovic. "Are your passwords safe: Energy-efficient bcrypt cracking with low-cost parallel hardware." 8th USENIX Workshop on Offensive Technologies (WOOT 14). 2014.

[2] Biryukov, Alex, Daniel Dinu, and Dmitry Khovratovich. "Argon2: new generation of memory-hard functions for password hashing and other applications." 2016 IEEE European Symposium on Security and Privacy (EuroS&P). IEEE, 2016.

Attackers in Server & DB - Session Stealing


- Session token generated by Google UUID (based on RFC 4122 and DCE 1.1: Authentication and Security Services)
- Valid for 180 days
- Stored in database as plain text
- Can be stolen and used to impersonate any user

```
mysql> select * from Sessions;
```

| Id | Token | CreateAt | ExpiresAt | LastActivityAt | UserId |
|----------------------------|-----------------------------|---------------|---------------|----------------|----------------------------|
| 1rz45jc6spyrpxnpx3q7a8e7ra | 3pqzoni6mtnezdxcnu7opnry6he | 1587621044790 | 1603173044790 | 1587621044790 | fg33amosfir67czezfkxgi57gc |

```
er":"Chrome/81.0.4044","csrf":"wftjtaqo37848f9ruok3msdd7o","is_guest":"false","os":"Mac OS","platform":"Macintosh"} |
```


Attackers in Server & DB - Integrated Applications



Ansible Playbook Installer

Ansible Playbook that installs a standalone version of Mattermost


by Tyler Tomlinson on November 2, 2015
Last updated on March 20, 2020
Programmed in Shell
License: MIT



Autolink Plugin

Automatically rewrite text matching a regular expression into a Markdown link


by Mattermost on June 14, 2018
Last updated on April 13, 2020
Programmed in Go
License: Apache 2.0



BotKube

Monitor & debug your Kubernetes cluster deployment, and receive best practices


by InfraCloud Technologies on April 4, 2019



Chess Plugin

Challenge a fellow Mattermost user to a game of Chess


by Matthew Dörner on March 2, 2020



Cron Monitoring

Receive alerts from Healthchecks cron job monitoring service to Mattermost


by Healthchecks on August 22, 2019
Last updated on April 14, 2020



Docker

Dockerfile for Mattermost in production


by Yi EungJun, Pan Luo, Kylene Pichou on December 14, 2016
Last updated on April 15, 2020
Programmed in Shell, Dockerfile
License: Apache 2.0



BigBlueButton Plugin

Start BigBlueButton video conference calls with screenshare in Mattermost


by Blindside Networks on July 5, 2018
Last updated on January 16, 2020
Programmed in Go, JavaScript
License: Apache 2.0



Bitbucket

Configurable, bidirectional app to integrate Mattermost and Bitbucket

by Herzum on April 24, 2019
Last updated on April 24, 2019
Programmed in N/A
License: N/A - not open source




GitHub Plugin

Subscribe to repositories, stay up-to-date with reviews, assignments and more

by Mattermost on August 9, 2018


Last updated on April 13, 2020
Programmed in Go
License: Apache 2.0



GitLab

Send events from GitLab to Mattermost through webhooks


by NetScout on December 19, 2016
Last updated on June 20, 2017
Programmed in Python, Shell
License: Apache 2.0



Jira Plugin

Send Jira ticket updates to Mattermost channels


by Mattermost on November 29, 2017
Last updated on April 14, 2020
Programmed in Go
License: Apache 2.0



GitLab Plugin

Subscribe to repositories, stay up-to-date with reviews, assignments and more


by Romain Maneschi on June 6, 2019
Last updated on April 13, 2020
Programmed in Go
License: Apache 2.0



Golang Bot Sample

Golang bot that listens to events and responds to messages in Mattermost

by Mattermost on July 4, 2016
Last updated on January 27, 2020
Programmed in Go
License: Apache 2.0

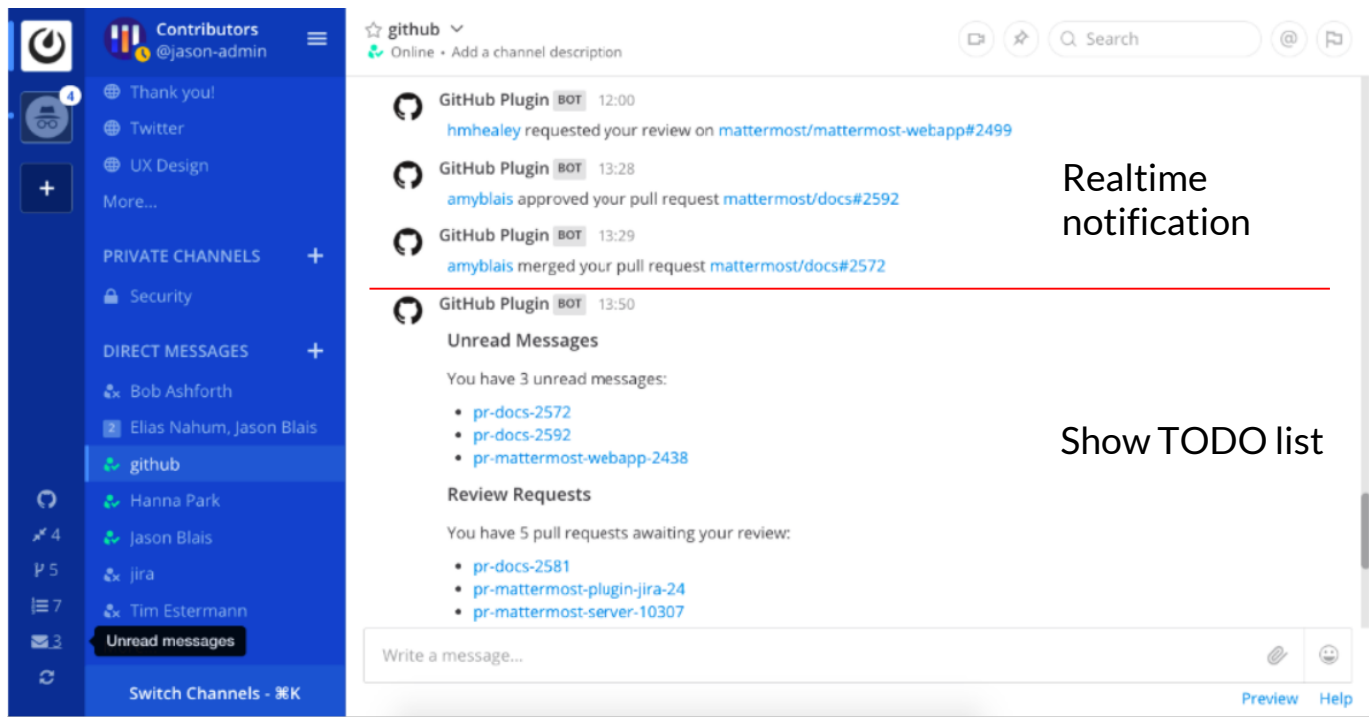


Mail2Most

Send emails to Mattermost with IMAP protocol support

by Carsten Seeger on June 30, 2019
Last updated on April 15, 2020
Programmed in Go
License: MIT

Attackers in Server & DB - Integrated Applications



The screenshot displays a Mattermost chat window. On the left is a sidebar with a list of channels and users. The 'github' channel is selected. The main chat area shows a series of messages from the 'GitHub Plugin BOT'. The messages include notifications about pull requests and merges. A red horizontal line separates a section of messages from the ones below. Below the line, there are two sections: 'Unread Messages' and 'Review Requests', both listing items with links. The 'Unread Messages' section lists three items: 'pr-docs-2572', 'pr-docs-2592', and 'pr-mattermost-webapp-2438'. The 'Review Requests' section lists three items: 'pr-docs-2581', 'pr-mattermost-plugin-jira-24', and 'pr-mattermost-server-10307'. At the bottom of the chat area is a text input field with the placeholder 'Write a message...'. To the right of the input field are icons for attachments and emojis. In the bottom right corner of the chat area are links for 'Preview' and 'Help'.

Contributors
@jason-admin

Thank you!
Twitter
UX Design
More...

PRIVATE CHANNELS +
Security

DIRECT MESSAGES +
Bob Ashforth
Elias Nahum, Jason Blais

github
Hanna Park
Jason Blais
Jira
Tim Estermann
Unread messages

Switch Channels - K

github
Online • Add a channel description

Search

GitHub Plugin BOT 12:00
hmhealey requested your review on [mattermost/mattermost-webapp#2499](#)

GitHub Plugin BOT 13:28
amyblais approved your pull request [mattermost/docs#2592](#)

GitHub Plugin BOT 13:29
amyblais merged your pull request [mattermost/docs#2572](#)

GitHub Plugin BOT 13:50

Unread Messages

You have 3 unread messages:

- [pr-docs-2572](#)
- [pr-docs-2592](#)
- [pr-mattermost-webapp-2438](#)

Review Requests

You have 5 pull requests awaiting your review:

- [pr-docs-2581](#)
- [pr-mattermost-plugin-jira-24](#)
- [pr-mattermost-server-10307](#)

Write a message...

Preview Help

Realtime notification

Show TODO list

Attackers in Server & DB - Integrated Applications



Mattermost GitHub Plugin by [redacted]
wants to access your [redacted] account



Notifications

Read access

This application will be able to read your notifications (no code access).

[? Learn more](#)



Repositories

Public repositories

This application will be able to **read and write all public repository data**. This includes the following:

- Code
- Issues
- Pull requests
- Wikis
- Settings
- Webhooks and services
- Deploy keys

[? Learn more](#)



Organizations and teams

Read-only access

This application will be able to read your organization, team membership, and private project boards.

[? Learn more](#)

Organization access

Attackers in Server & DB - Integrated Applications



Mattermost GitHub Plugin by [redacted]
wants to access your [redacted] account



Notifications



Organizations and teams

Read-only access

This application will be able to read your organization, team membership, and private project boards.



Authorizing OAuth Apps - Git



help.github.com/en/github/authenticating-to-github/authorizing-oauth-apps

Keeping your account and data
secure

Note: Currently, you can't scope source code access to read-only.

This application will be able to read and write all public repository data. This includes the following:


Enable Code Previews

☐ true ☒ false

(Optional) Allow the plugin to expand permalinks to github files with an actual preview of the linked file.

- Webhooks and services
- Deploy keys

[Learn more](#)



Attackers in Server & DB - Integrated Applications

- GitHub token is AES-256 encrypted and stored in database
- Encryption key is randomly generated and stored in a json file
- Still safe when the database is dumped, but not safe when fully compromised

Attackers in Server & DB - Integrated Applications

Mattermost Configuration

Step 1: Register an OAuth

1. Go to <https://gitlab.com/projects>
2. Set the following values:
 - **Name:** Mattermost GitLab Plugin (your company name)
 - **Redirect URI:** <https://your-mattermost-url.com/plugins/com.gitlab.mattermost> replacing <https://your-mattermost-url.com> with your Mattermost URL
3. Select **api** and **read_user** in **Scopes**

Scopes

☐ api

Grants complete read/write access to the API, including all groups and projects, the container registry, and the package registry.

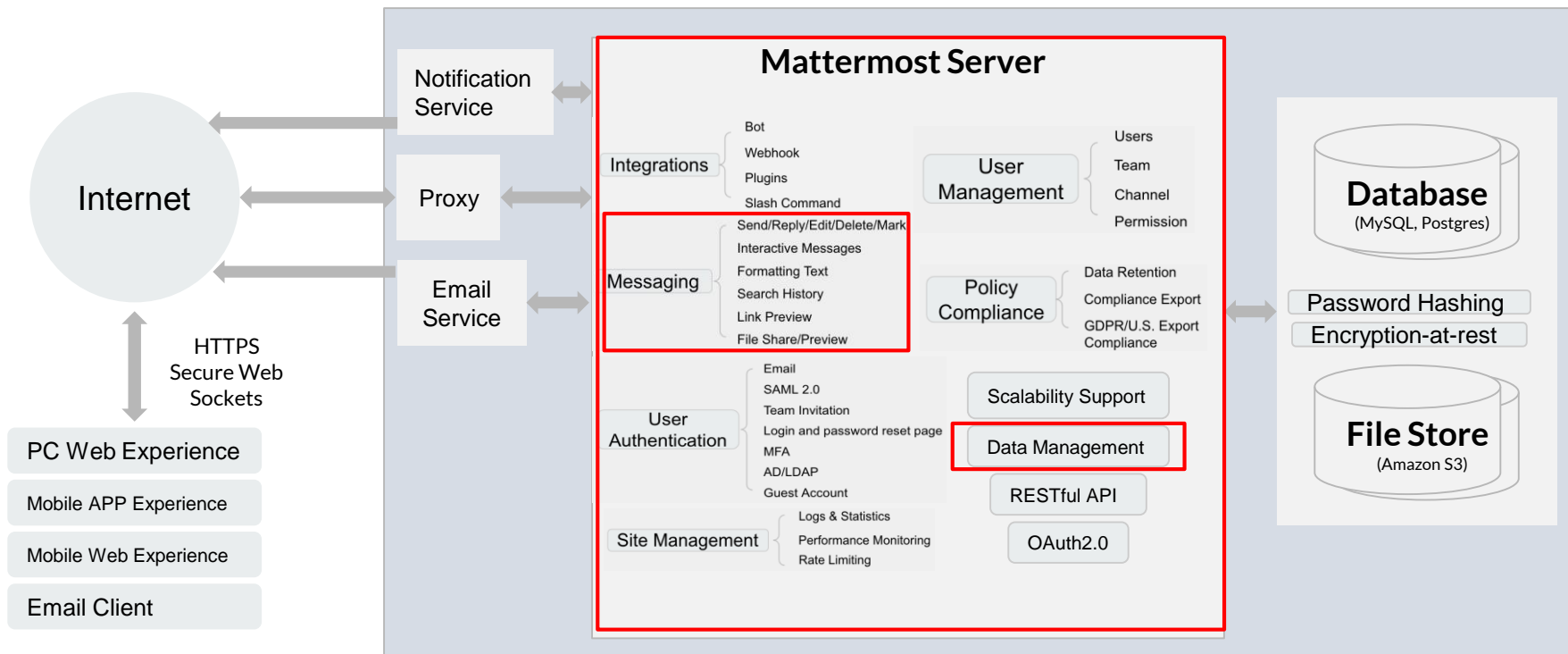
☐ read_user

Grants read-only access to the authenticated user's profile through the /user API endpoint, which includes username, public email, and full name. Also grants access to read-only API endpoints under /users.

☐ read_api

Grants read access to the API, including all groups and projects, the container registry, and the package registry.

Attackers as Non-Admin Mattermost Users





Attackers as Non-Admin Mattermost Users

A solid step. But what's next?

Target

- Intellectual property and trade secret (Posts), as the threat model suggests

Approach

- Become Mattermost's system admin (**XSS attack**, password cracking, SQL injection etc.)
- Trick other users (Phishing)
- Dump the database (SQL injection)



Attackers as Non-Admin Users - XSS attack

Session token is stored in cookies

~~Steal token by XSS attack - document.cookie;~~ Token is set to be **httponly**

RESTful API of role management

~~Send <script>XMLHttpRequest.open()</script> to the admin~~ <> will be escaped to > <

~~Send [click me](JavaScRipt:alert();) to the admin~~ URL will be checked before rendered

Attackers as Non-Admin Users - XSS attack

Session token is stored in cookies

~~Steal token by XSS attack - document.cookie; Token is set to be httponly~~



weihaio 1:58 PM

```
<script>alert("hacked");</script>
```

~~Send <script>XMLHttpRequest.open()</script> to the admin~~ `<>` will be escaped to `<` `>`;

	Headers	Preview	Response	Initiator	Timing	Cookies
1	:	:	"", "message": "\u003cscript\u003ealert(\"hacked\");\u003c/script\u003e", "			

URL will be checked before rendered

```
<p>&lt;script&gt;alert("hacked")  
&lt;/script&gt;</p>
```

Attackers as Non-Admin Users - XSS attack

Session token is stored in cookies

~~Steal token by XSS attack - document.cookie; Token is s~~

[click me](javascript:alert();)
↓
click me

RESTful API of role management

~~Send <script>XMLHttpRequest.open()</script> to the admin~~ <> will be escaped to > <

~~Send [click me](JavaScripT.alert();) to the admin~~ URL will be checked before rendered



Attackers as Non-Admin Users - XSS attack

Backend sends unescaped text file to user for file preview

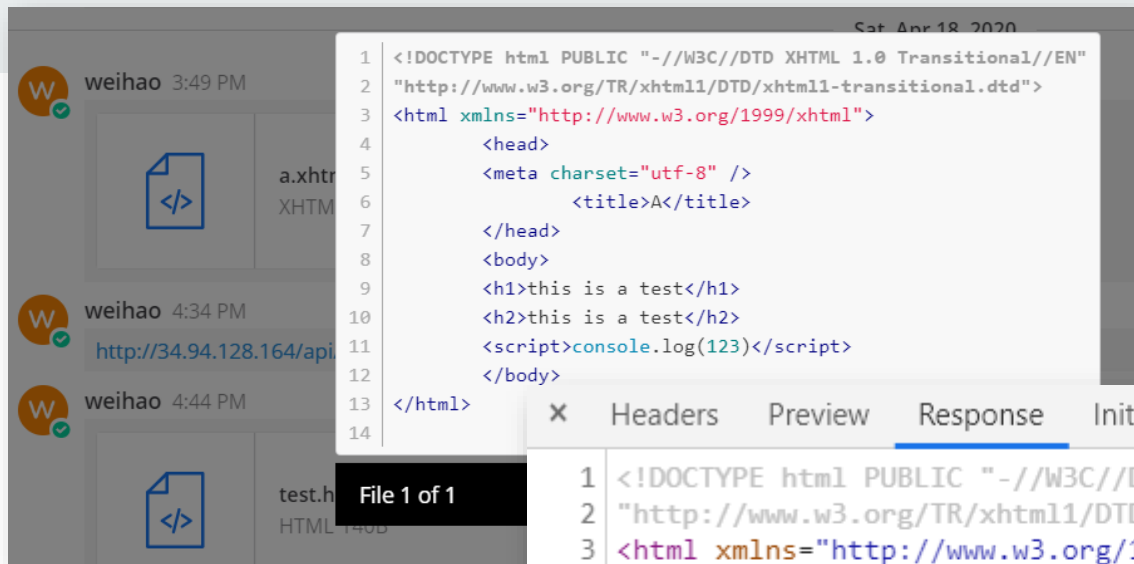
~~Upload a malicious HTML file and trick the admin to preview it~~

Frontend escapes it

File preview won't be triggered for a link to the file

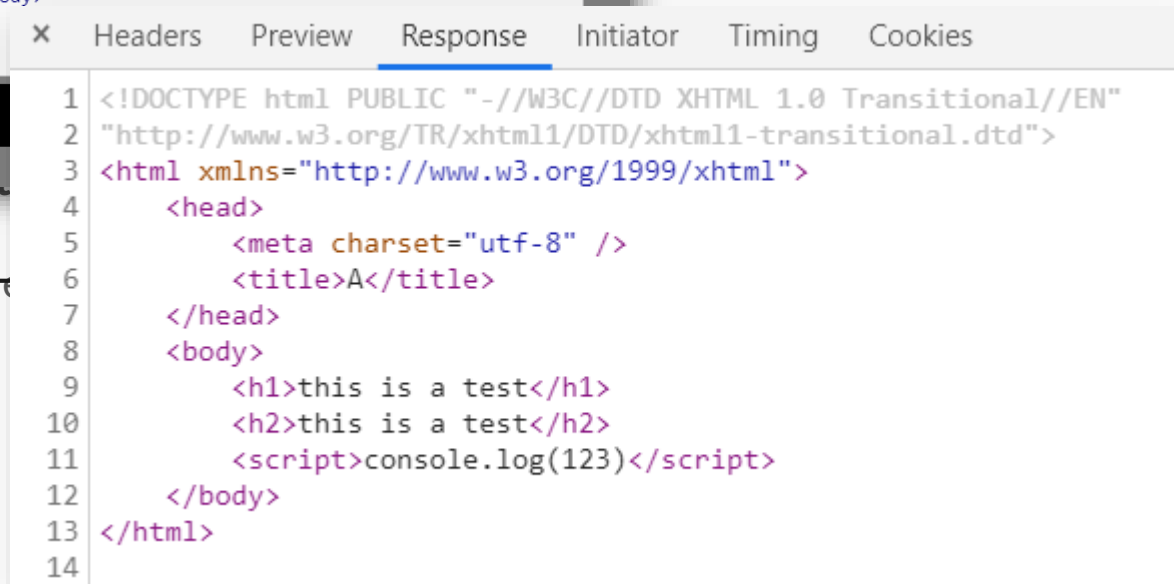
~~Bypass the frontend escape by posting a link to the HTML file~~

Content-Disposition in the header is set to attachment



- XSS attack

Preview



es it

in the
chment

Bypass the frontend

- XSS attack

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml">
4   <head>
5     <meta charset="utf-8" />
6     <title>A</title>
7   </head>
8   <body>
9     <h1>this is a test</h1>
10    <h2>this is a test</h2>
```



weihao 4:44 PM



test.html

HTML 140B



<http://34.94.128.164/api/v4/files/pt7fi1fwxbyzxq4uqik9jf95kw?download=0>

<http://34.94.128.164/api/v4/files/pt7fi1fwxbyzxq4uqik9jf95kw>

```
11 </script>console.log(123)</script>
12 </body>
13 </html>
14
```

- XSS attack

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml">
4   <head>
5     <meta charset="utf-8" />
6     <title>A</title>
7   </head>
8   <body>
9     <h1>this is a test</h1>
10    <h2>this is a test</h2>
```



weihao 4:44 PM



test.html

Content-Security-Policy: frame-ancestors 'self'; script-src 'self' cdn.segment.com/analytics.js/

Bypass the

<http://34.94.128.164/api/v4/files/pt7fi1fwxbyzxq4uqik9jf95kw?download=0>

<http://34.94.128.164/api/v4/files/pt7fi1fwxbyzxq4uqik9jf95kw>

```
11 <script>console.log(123)</script>
12 </body>
13 </html>
14
```

Attackers as Non-Admin Users - Phishing



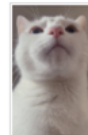
wei hao 1:55 PM

<https://twitter.com/PebblesPuss2014/status/1251476543538331648>

Twitter

Princess Pebbles on Twitter

"Have a safe #Caturday everyone 🐾"



1. Post a link



2. Send a GET request

3. Return full content

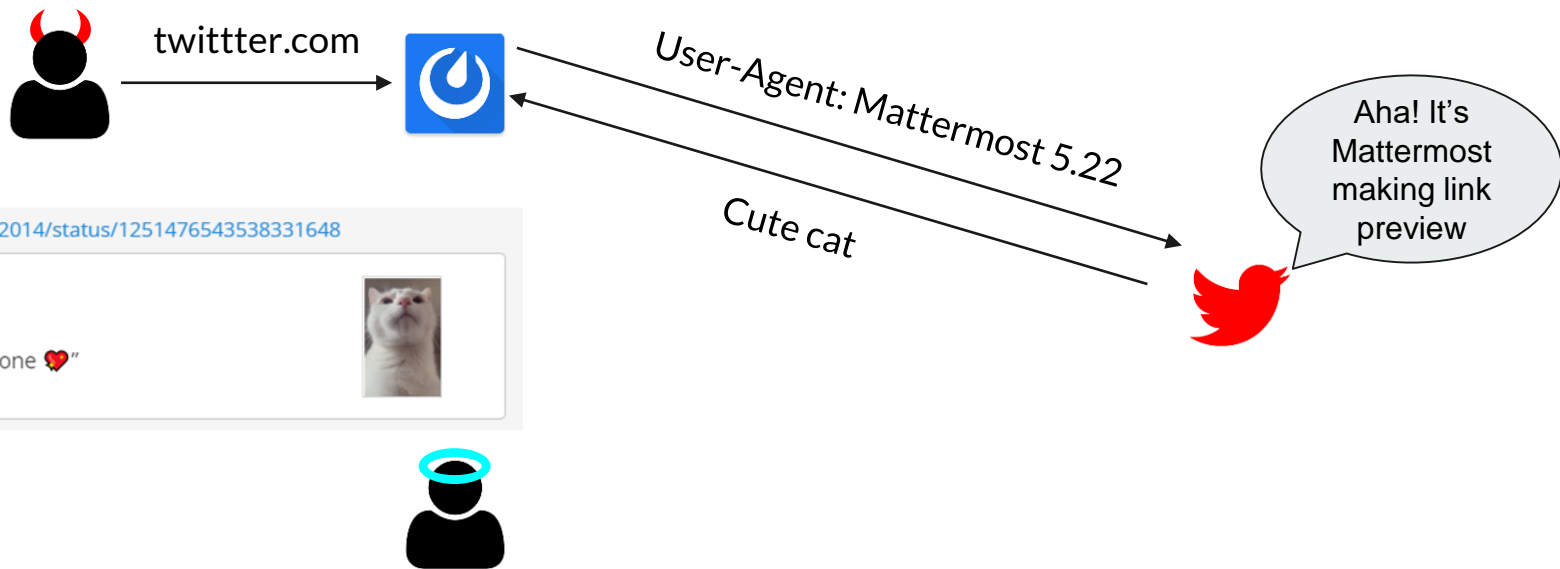


4. Summarize the content and display

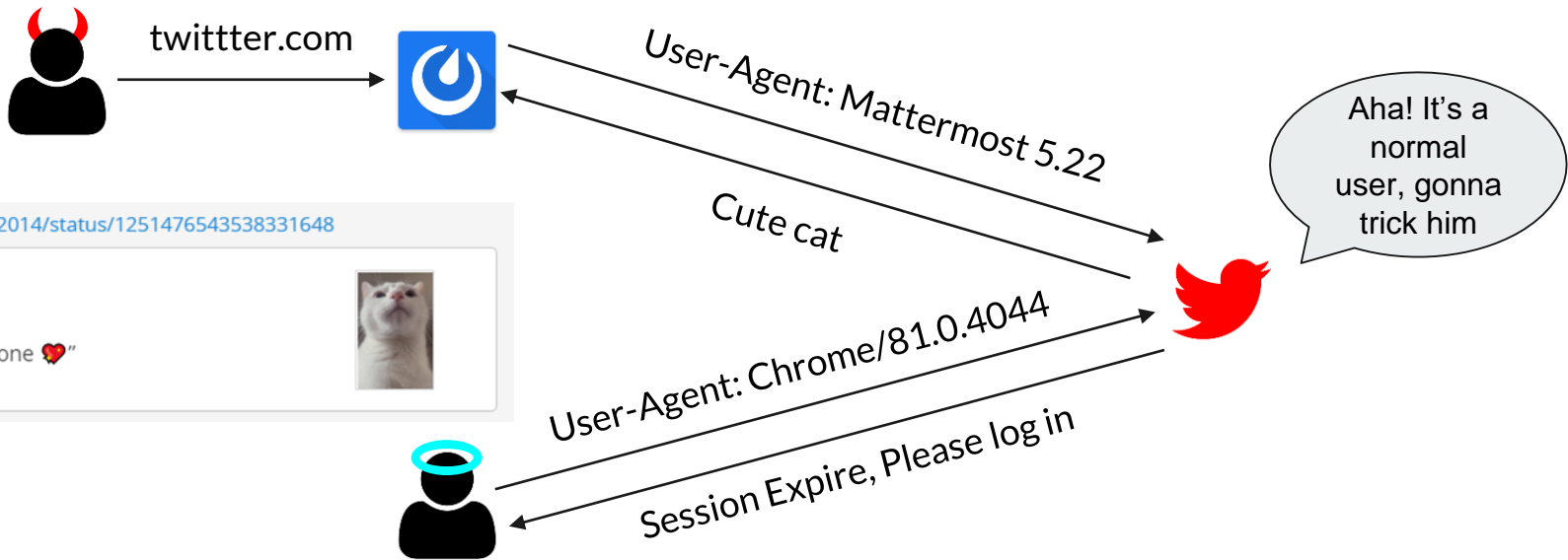


User-Agent: Mattermost 5.22

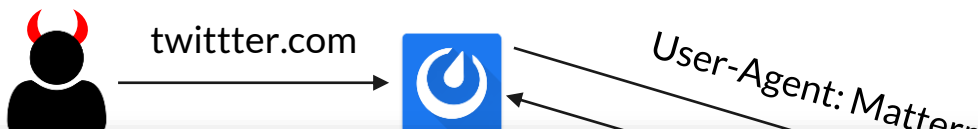
Attackers as Non-Admin Users - Phishing



Attackers as Non-Admin Users - Phishing



Attackers as Non-Admin Users - Phishing



wei hao 1:55 PM

<https://twitter.com/PebblesPuss2014/status/1251476543538331648>

Twitter

Princess Pebbles on Twitter

"Have a safe #Caturday everyone 🍷"



Session expired, please log in again

Username Password



Session Expire, Plea...

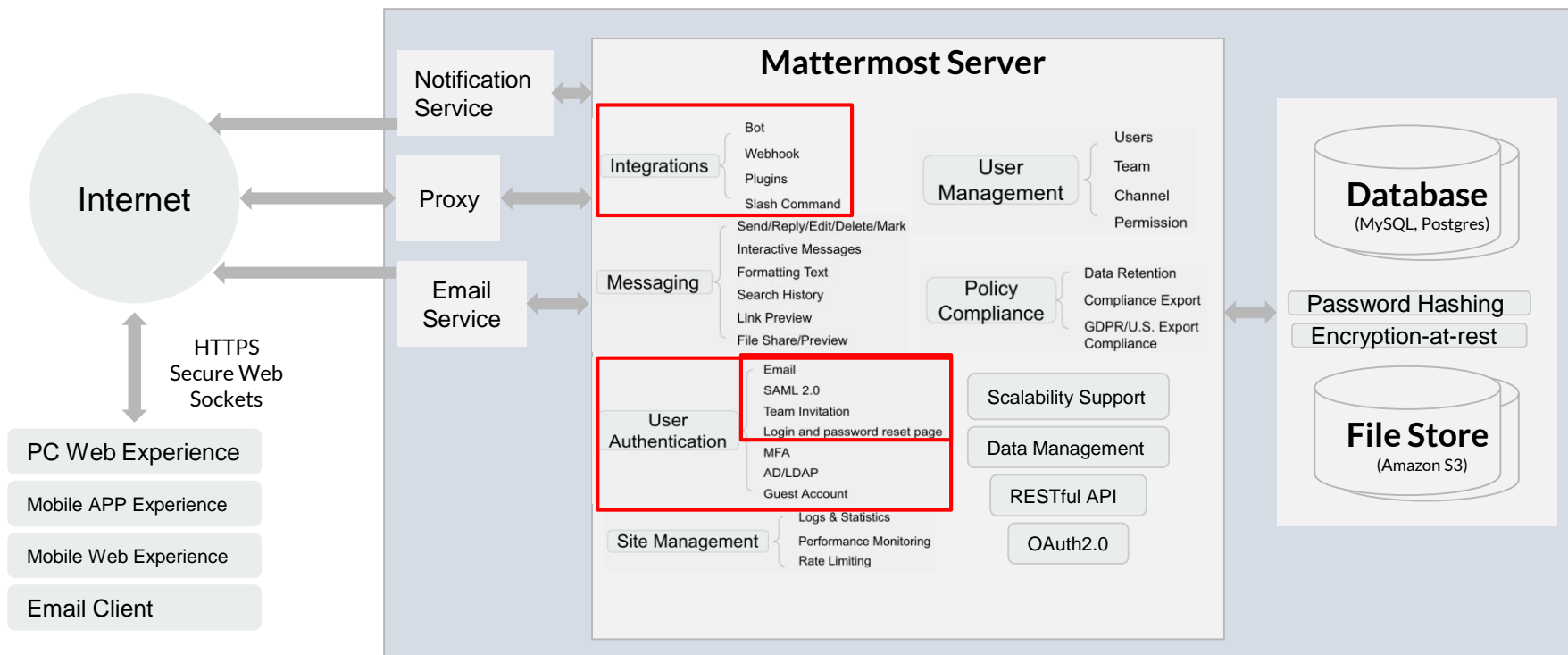
Attackers as Non-Admin Users - SQL Injection

```
func query(e SqlExecutor, query string, args ...interface{}) (*sql.Rows, error) {
    executor, ctx := extractExecutorAndContext(e)

    if ctx != nil {
        return executor.QueryContext(ctx, query, args...)
    }

    return executor.Query(query, args...)
}
```


Attackers as Non-Users





Attackers as Non-Users - User Onboard

- Email
- SAML
- Team Invitation

User Onboard - Email-based Registration

Signup

Enable Account Creation:

☒ true ☐ false

When false, the ability to create accounts is disabled. The create account button displays error when pressed.

Restrict account creation to specified email domains:

berkeley.edu, gmail.com, example.com

User accounts can only be created from a specific domain (e.g. "mattermost.org") or list of comma-separated domains (e.g. "corp.mattermost.com, mattermost.org"). This setting only affects email login for users. For Guest users, please add domains under Signup > Guest Access.

Enable Open Server:

☒ true ☐ false

When true, anyone can sign up for a user account on this server without the need to be invited.

Enable Email Invitations:

☐ true ☒ false

When true users can invite others to the system using email.

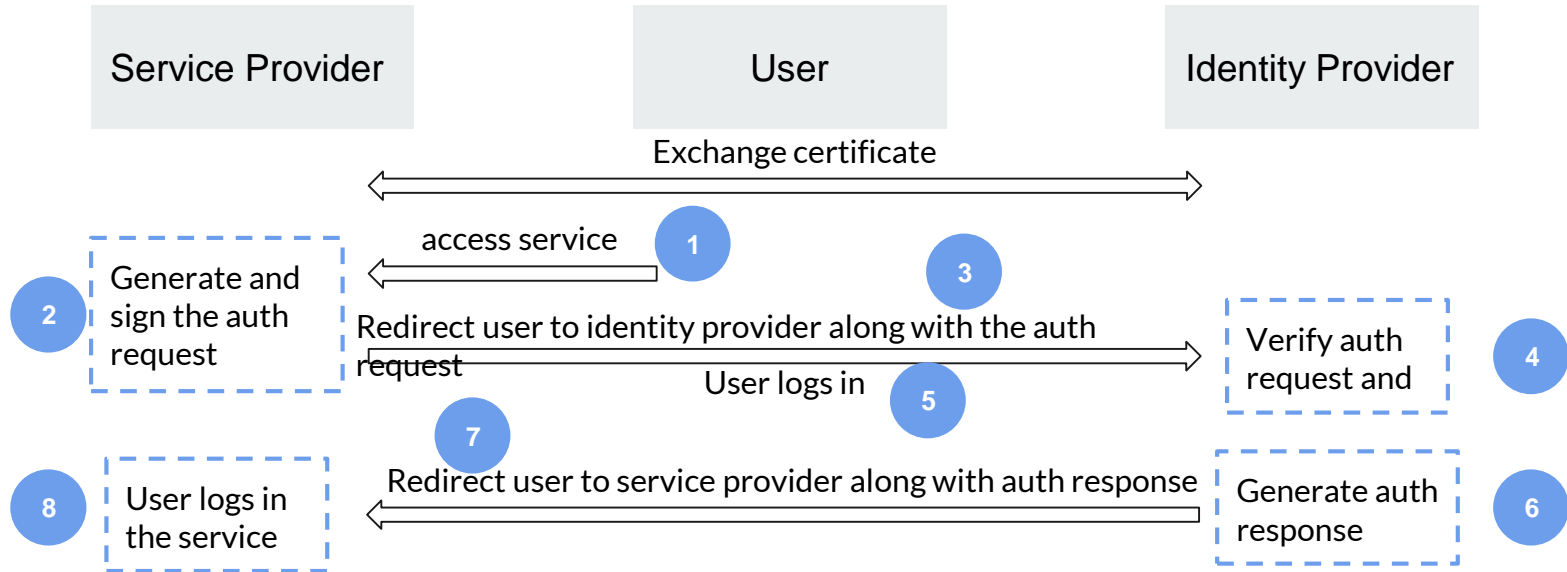
Invalidate pending email invites



User Onboard - SAML-based Registration

- A XML-based protocol for **exchanging identities between Identity Provider and Service Provider**
- Used for Authentication
- Commonly used in Single-sign On application
- Fundamentally different from OAuth 2.0

How SAML 2.0 Works



SAML 2.0 - Sample Auth Request

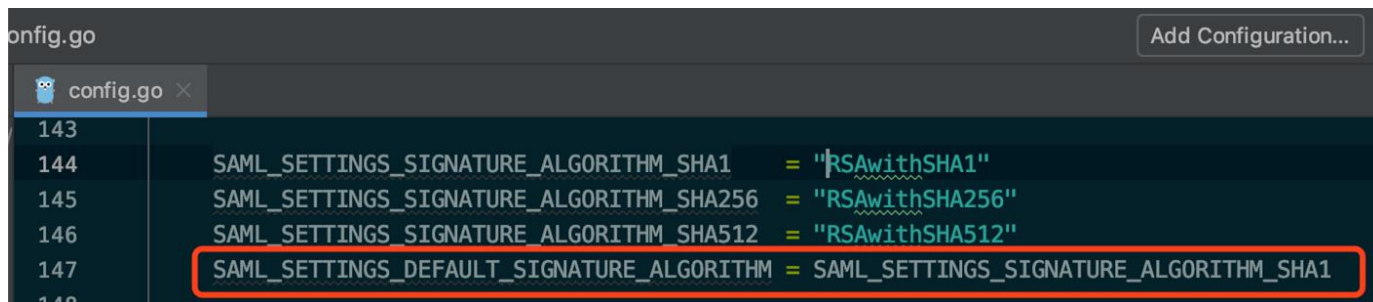
```
<samlp:AuthnRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
  xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
  ID="pfx41d8ef22-e612-8c50-9960-1b16f15741b3"
  Version="2.0"
  ProviderName="SP_test"
  IssueInstant="2014-07-16T23:52:45Z"
  Destination="http://idp.example.com/SSOService.php"
  ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
  AssertionConsumerServiceURL="http://sp.example.com/demo1/index.php?acs">

  <saml:Issuer>http://sp.example.com/demo1/metadata.php</saml:Issuer>
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    ...
  </ds:Signature>
  <samlp:NameIDPolicy Format="urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress" AllowCreate="true"/>
  <samlp:RequestedAuthnContext Comparison="exact">
    <saml:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport</saml:AuthnContextClassRef>
  </samlp:RequestedAuthnContext>
</samlp:AuthnRequest>
```

SAML 2.0 - Sample Auth Response

```
<samlp:Response xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
  xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
  ID="_8e8dc5f69a98cc4c1ff3427e5ce34606fd672f91e6"
  Version="2.0"
  IssueInstant="2014-07-17T01:01:48Z"
  Destination="http://sp.example.com/demo1/index.php?acs"
  InResponseTo="ONELOGIN_4fee3b046395c4e751011e97f8900b5273d56685">
  <saml:Issuer>http://idp.example.com/metadata.php</saml:Issuer>
  <saml:Assertion
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    ID="pfx61086f73-67d9-9633-17c5-f36349c000c8"
    Version="2.0"
    IssueInstant="2014-07-17T01:01:48Z">
    <saml:Issuer>http://idp.example.com/metadata.php</saml:Issuer><ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    ...
    <saml:AttributeStatement>
      <saml:Attribute Name="uid" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">
        <saml:AttributeValue xsi:type="xs:string">test</saml:AttributeValue>
      </saml:Attribute>
      <saml:Attribute Name="mail" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">
        <saml:AttributeValue xsi:type="xs:string">test@example.com</saml:AttributeValue>
      </saml:Attribute>
    </saml:AttributeStatement>
  </saml:Assertion>
</samlp:Response>
```

SAML Signature Algorithm is not Safe



```
config.go
Add Configuration...
config.go x
143
144 SAML_SETTINGS_SIGNATURE_ALGORITHM_SHA1 = "RSawithSHA1"
145 SAML_SETTINGS_SIGNATURE_ALGORITHM_SHA256 = "RSawithSHA256"
146 SAML_SETTINGS_SIGNATURE_ALGORITHM_SHA512 = "RSawithSHA512"
147 SAML_SETTINGS_DEFAULT_SIGNATURE_ALGORITHM = SAML_SETTINGS_SIGNATURE_ALGORITHM_SHA1
148
```

SHA-1 is a Shambles

First Chosen-Prefix Collision on SHA-1
and Application to the PGP Web of Trust

Gaëtan Leurent¹ and Thomas Peyrin^{2,3}

¹ Inria, France

² Nanyang Technological University, Singapore

³ Temasek Laboratories, Singapore

The first collision for full SHA-1

¹, Elie Bursztein², Pierre Karpman¹, Ange Albertini², Ya

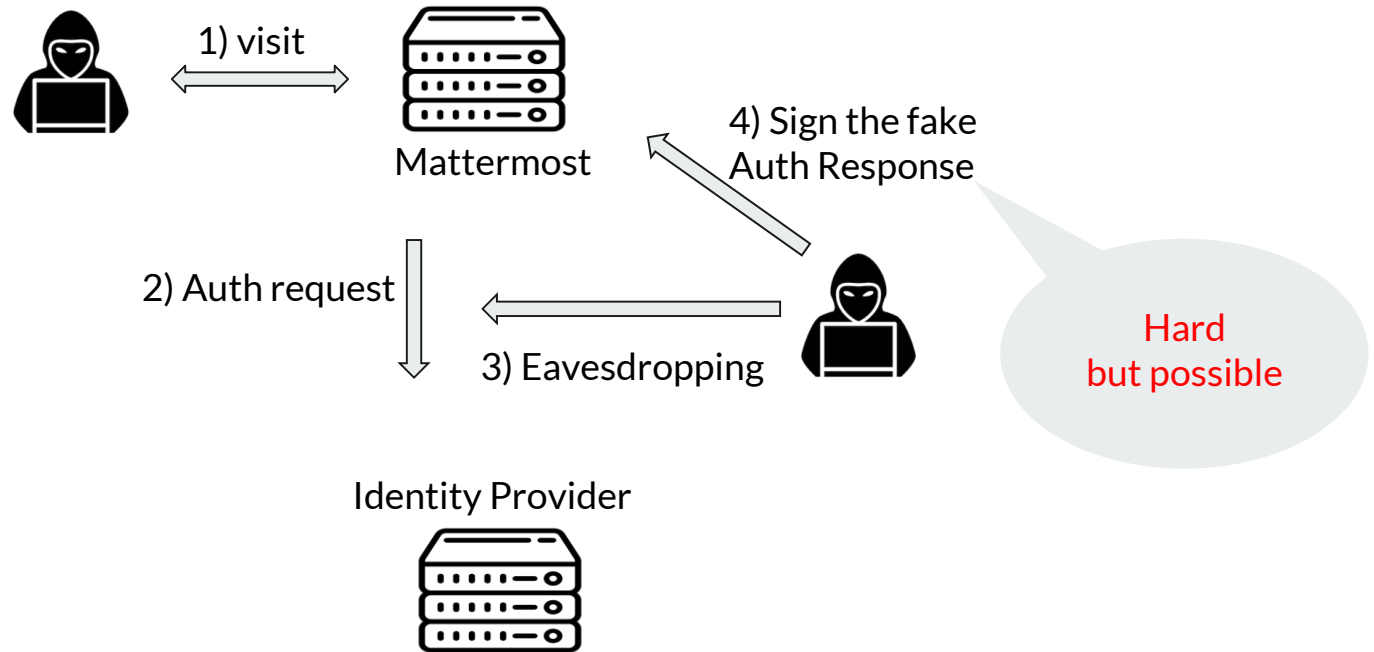
¹ CWI Amsterdam

² Google Research

info@shattered.io

<https://shattered.io>

If attackers could forge the Signature



User Onboard - Team Invitation

Signup

Enable Account Creation:

☒ true ☐ false

When false, the ability to create accounts is disabled. The create account button displays error when pressed.

Restrict account creation to specified email domains:

berkeley.edu, gmail.com, example.com

User accounts can only be created from a specific domain (e.g. "mattermost.org") or list of comma-separated domains (e.g. "corp.mattermost.com, mattermost.org"). This setting only affects email login for users. For Guest users, please add domains under Signup > Guest Access.

Enable Open Server:

☒ true ☐ false

When true, anyone can signup for a user account on this server without the need to be invited.

Enable Email Invitations:

☐ true ☒ false

When true users can invite others to the system using email.

Invalidate pending email invites

User Onboard - Team Invitation



Invite Members to 261test

http just for test

Share This Link

http://34.94.128.164/signup_user_complete/?id=pqqqcbcx9iprqpmjhrf

 Copy Link

Share this link to invite people to this team.



Attackers as Non-Users - Team Invitation

uuid build passing

The uuid package generates and inspects UUIDs based on [RFC 4122](#) and DCE 1.1: Authentication and Security Services.

This package is based on the [github.com/pborman/uuid](#) package (previously named [code.google.com/p/go-uuid](#)). It differs from these earlier packages in that a UUID is a 16 byte array rather than a byte slice. One loss due to this change is the ability to represent an invalid UUID (vs a NIL UUID).

Install

```
go get github.com/google/uuid
```

Attackers as Non-Users - Team Invitation



Invite Members to 261test

Share This Link

http://34.94.128.164/signup_user_complete/?id=pqqcbcx9iprqpmjhrl

[Copy Link](#)

Share this link to invite people to this team.



Invite Members to 261test

Share This Link

http://34.94.128.164/signup_user_complete/?id=pqqcbcx9iprqpmjhrl

[Copy Link](#)

Attackers as Non-Users - Team Invitation

```
mysql> select * from Teams;
```

Id	CreateAt	UpdateAt	DeleteAt	DisplayName	Name	Description	Email
InviteId	AllowOpenInvite	LastTeamIconUpdate	SchemeId	GroupConstrained			
upg3hj34kt8czngdgfxzfgj4gh	1587095073047	1587095073047	0		moweihao		changzecu
z98m7tmpbf8d748sn3pt7gty	0	0	NULL		NULL		
w9wpbiw13tg18p31ha4rusbpoc	1586230140033	1586230140033	0	261test	test261		weihao_dor
pqqcbcx9ipraqmjhrf1nha3bfa	0	0	NULL		NULL		



Attackers as Non-Users - Login

⚠ Enter a valid email or username and/or password, or sign in using another method.

Sign in

Password Reset

If the account exists, a password reset email will be sent to:
admin@test.com

Please check your inbox.



Attackers as Non-Users - Login

⚠ Your account is locked because of too many failed password attempts. Please reset your password.

changze

...

Sign in

Attackers as Non-Users - Login

⚠ Your account is locked because of too many failed password attempts. Please reset your password.

changze

...

Sign in

⚠ Enter a valid email or username and/or password, or sign in using another method.

changzeeeeeeeee

...

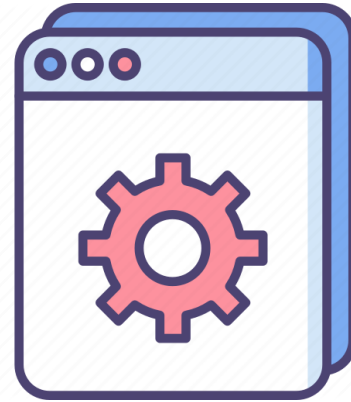
Sign in

Attackers as Non-Users - Phishing with webhook



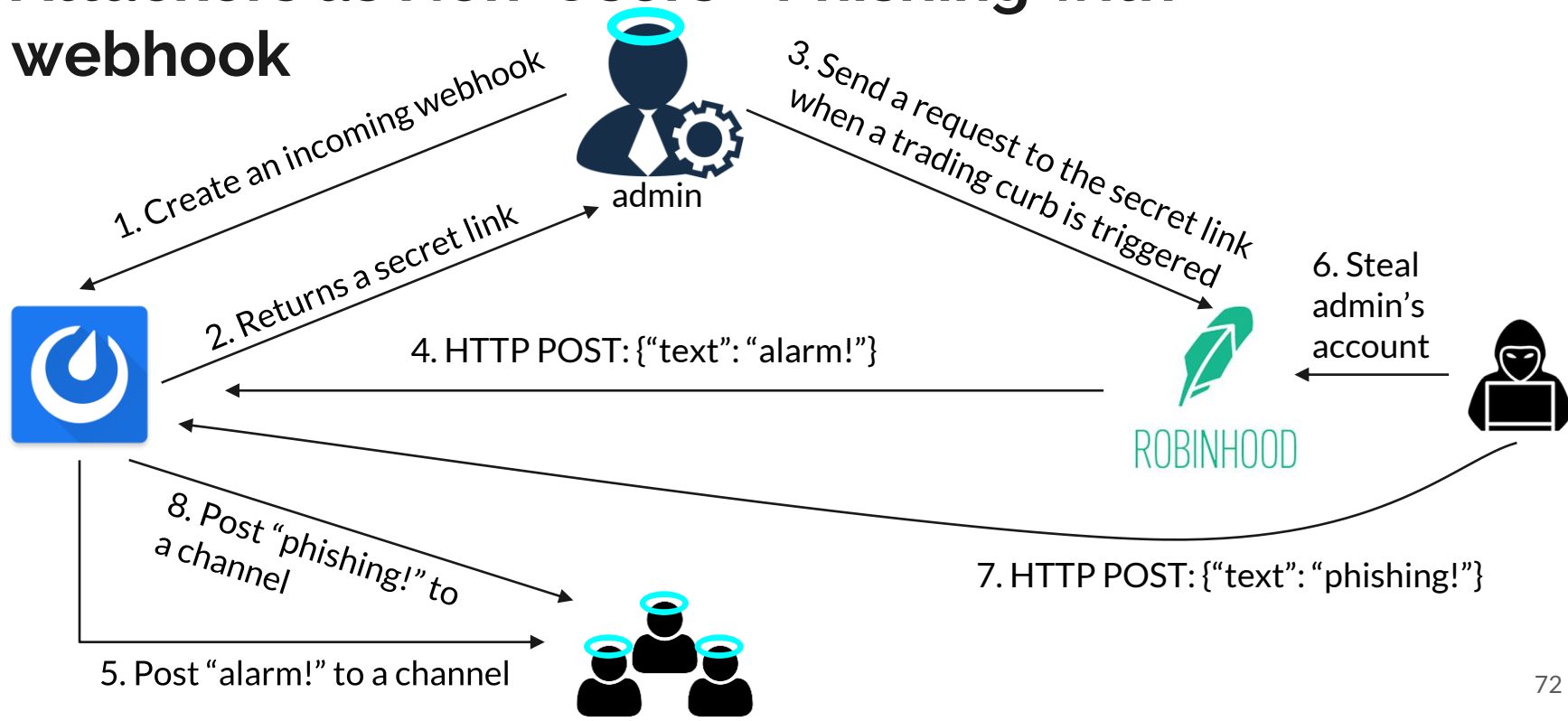
Application A

Exposed URL



Application B

Attackers as Non-Users - Phishing with webhook



Attackers as Non-Users - Phishing with webhook

- A BOT tag is attached to messages posted by webhook



weihao BOT 4:51 PM

Changze is our new team leader now, please add changze@evil.com to the github repo.

- Allows username and icon override



Robinhood BOT 4:51 PM

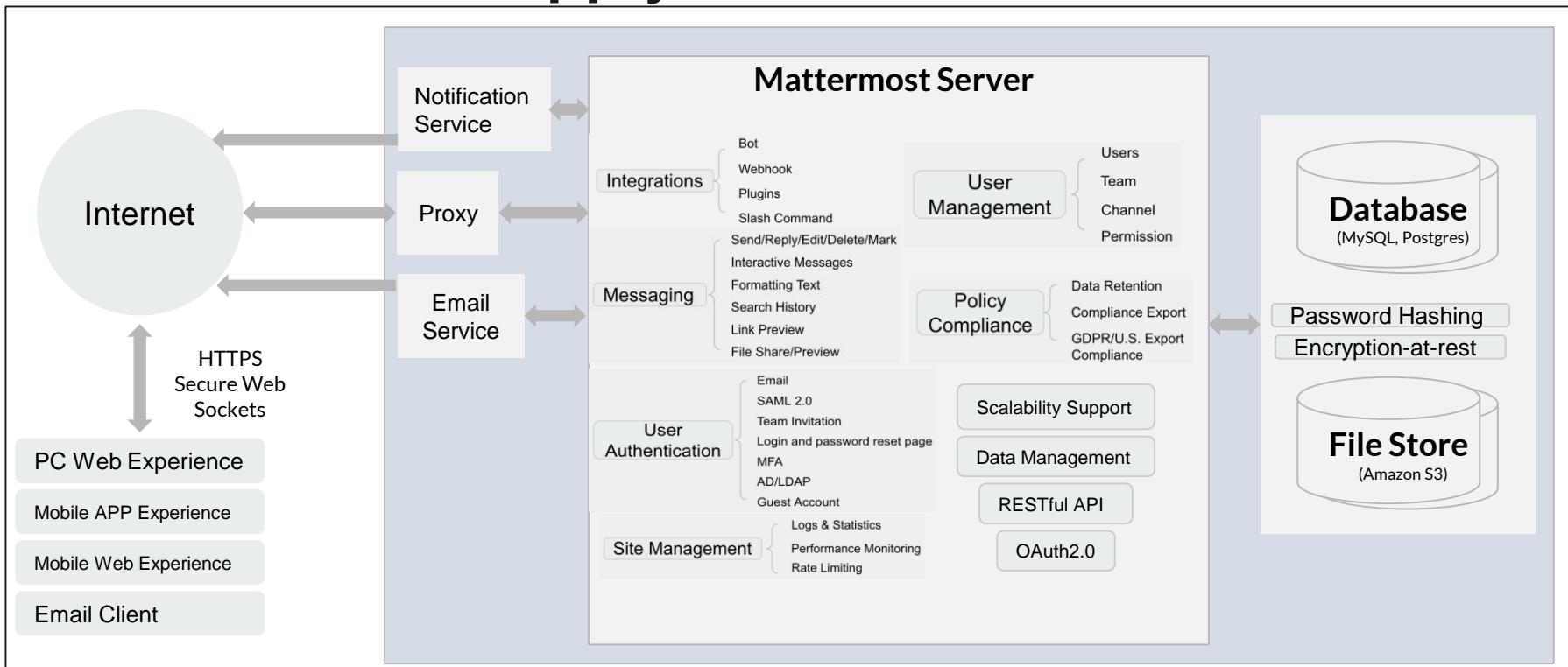
Your robinhood account is logged out, please [login](#) again. 🎉

Enable integrations to override
usernames:

☐ true ☒ false

When true, webhooks, slash commands and other integrations, such as [Zapier](#), will be allowed to change the username they are posting as. Note: Combined with allowing integrations to override profile picture icons, users may be [able to perform phishing attacks](#) by attempting to impersonate other users.

Attackers in Supply Chain Vulnerabilities



Attackers in Supply Chain Vulnerabilities

49 personal projects

Mattermost Server 271 open source projects in total

The screenshot displays a grid of GitHub repository cards for various Go projects. Each card includes the repository name, a 'Code' button, and a summary of issues and pull requests. The projects are arranged in three rows. The bottom row is highlighted with a red border.

Repository	Issues	Pull requests
Masterminds / squirrel	7	
nytimes / gziphandler	12	
armon / go-metrics	20	
beevik / etree	4	
miekg / dns		
blang / semver	14	7
disintegration / imaging	6	0
fnotify / fnotify	120	14
go-asn1-ber / asn1-ber	2	4
go-sql-driver / mysql	48	
gorilla / mux	11	
hako / durafmt	1	
hashicorp / go-immutable-radix	2	2
hashicorp / go-plugin	11	
sirupsen / logrus	80	
dgryski / dgoogauth	2	
jaytaylor / html2text	4	3
jmoiron / sqlx	178	53
muesli / smartcrop	4	2
sean- / seed	0	0

Attackers in Supply Chain Vulnerabilities

 jaytaylor / [html2text](#)

Watch ▾

10

★ Star

273

<> Code

! Issues 4

🔗 Pull requests 3

▶ Actions

📁 Projects 0

📖 Wiki

🛡 Security 0

📊 Insights

Golang HTML to plaintext conversion library <https://jaytaylor.com/html2text>

go

golang

html2text

html-emails

plaintext

```
func SendMail(c smtpClient, mail mailData, fileBackend filesstore.FileBackend, date time.Time) *model.AppError {
    mlog.Debug("sending mail", mlog.String(key: "to", mail.smtpTo), mlog.String(key: "subject", mail.subject))

    htmlMessage := "\r\n<html><body>" + mail.htmlBody + "</body></html>"

    txtBody, err := html2text.FromString(mail.htmlBody)
    if err != nil {
        mlog.Warn("Unable to convert html body to text", mlog.Err(err))
        txtBody = ""
    }
}
```



Attackers Aiming at Plugins

- Allows installing plugin binaries from untrusted third party
- No permission control for plugins
- Plugins have the same permission as Mattermost itself

Security

Plugins are intentionally powerful and not artificially sandboxed in any way and effectively become part of the Mattermost server. Server plugins can execute arbitrary code alongside your server and webapp plugins can deploy arbitrary code in client browsers.

While this power enables deep customization and integration, it can be abused in the wrong hands. Plugins have full access to your server configuration and thus also to your Mattermost database. Plugins can read any message in any channel, or perform any action on behalf of any user in the webapp.

You should only install custom plugins from sources you trust to avoid compromising the security of your installation.



Future Work Beyond Our Project

- A closer look at Mattermost
 - Notification system
 - Client side software
 - Upgrading
 - License Management
- Apply our investigation method to similar applications (e.g., Slack)



Thanks