7 лучших API сканирования безопасности для обнаружения рисков на сайте

10.05.2024

В Сети полно вредоносных страниц. К сожалению, они могут существовать и на сайтах ваших клиентов/поставщиков. Сегодня ни один бизнес не обходится без интеграции, которая питает или обеспечивает вход на сайт клиента или поставщика. Конечно, ваш бизнес не будет существовать без этих служб, но иногда он представляет угрозу именно *из-за* них. Внешние сайты, с которыми вы взаимодействуете, могут содержать вредоносный контент (установленный специально или скомпрометированный третьей стороной), и если этот контент попадет в нужное место, последствия могут быть катастрофическими.

Нельзя ли проверять сайты на наличие вредоносных страниц вручную? Казалось бы, грамотный разработчик должен

уметь сканировать страницы на наличие уязвимостей. К сожалению, это даже близко не похоже на реальность по многим причинам:

- Разработчики не специализируются на обнаружении/безопасности. Их опыт заключается в создании сложного программного обеспечения путем объединения множества мелких подсистем; иными словами, у них просто нет таких навыков.
- Даже если вам попадется достаточно талантливый разработчик, задача окажется непосильной. Типичная многофункциональная веб-страница содержит тысячи строк кода — сшить их вместе, чтобы проработать общую картину, а также мелкие лазейки, — не что иное, как кошмар. С таким же успехом можно приказать кому-нибудь съесть на обед целого слона!
- Чтобы уменьшить время загрузки страниц, сайты часто сжимают и минифицируют свои CSS и JavaScript-файлы. В результате код превращается в такую кашу, что его совершенно невозможно прочитать.

Before

```
$(document).ready(function () {

    // COLLAPSABLE WIDGETS (on Dashboard? & Profile pages)
    // toggle widget box contents
    $('a.toggle_box_contents').bind('click', toggleContent);

    // WIDGET GALLERY EDIT PANEL
    // Sortable widgets
    var els = ['#leftcolumn_widgets', '#middlecolumn_widgets', '#rightcolumn_widgets', '#widget_picker_gallery' ];
    var $els = $(els.toString());

    $els.sortable({
        items: '.draggable_widget',
        handle: '.drag_handle',
        forcePlaceholderSize: true,
        placeholder: 'ui-state-highlight',
```

After

```
$(document).ready(function(){$('a.toggle_box_contents').bind('click',toggleContent);var els=['#leftcolumn_widgets','#middlecolumn_widgets','#rightco
function elgg_slide_toggle(activateLink,parentElement,toggleElement){$(activateLink).closest(parentElement).find(toggleElement).animate({"height":"t
function outputWidgetList(forElement){return($("input[name='handler'], input[name='guid']",forElement).makeDelimitedList("value"));}
jQuery.fn.makeDelimitedList=function(elementAttribute){var delimitedListArray=new Array();var listDelimiter="::";this.each(function(e){var listEleme
function widget_state(forWidget){var thisWidgetState=$.cookie(forWidget);if(thisWidgetState=='collapsed'){forWidget="#"+forWidget;$(forWidget).find(
var toggleContent=function(e){var targetContent=$('div.collapsable_box_content',this.parentNode.parentNode);if(targetContent.css('display')=='none')
return false;};function widget_moreinfo(){$("img.more_info").hover(function(e){var widgetdescription=$("input[name='description']",this.parentNode.p
else{$("#widget_moreinfo").css("top",(e.pageY+10)+"px").css("left",(e.pageX-210)+"px").fadeIn("medium");}},function(){$("#widget_moreinfo").remove()
var expires='';if(options.expires&&(typeof options.expires=='number'||options.expires.toUTCString)){var date;if(typeof options.expires=='number'){da
expires='; expires='+date.toUTCString();}
var path=options.path?'; path='+(options.path):'';var domain=options.domain?'; domain='+(options.domain):'';var secure=options.secure?'; secure':'';
return cookieValue;}};$.fn.elgg_dropdownmenu=function(options){options=$.extend({speed:350},options||{});this.each(function(){var root=this,zIndex=5
function getActuator(ele){if(ele.nodeName.toLowerCase()=='ul'){return $(ele).parents('li')[0];}else{return ele;}}
function hide(){var subnav=getSubnav(this);if(!subnav)return;$.data(subnav,'cancelHide',false);setTimeout(function(){if(!$.data(subnav,'cancelHide')
```

Как вы думаете, что делает этот код?

Если это все еще выглядит читабельно, то это потому, что добрые души решили сохранить имена переменных в большом контексте. Попробуйте взять исходный код jQuery, который кто-

то может разместить на своем сайте и подделать (две строки где-то внизу этого беспорядка):

Не говоря уже о том, что в исходниках почти 5 000 строк кода. Речь идет только об одном скрипте. На веб-странице обычно размещается 5-15 скриптов, и вполне вероятно, что вы работаете с 10-20 веб-страницами в общей сложности. Представьте, что вам придется делать это каждый день… Или, что еще хуже, несколько раз в день! К счастью, можно быстро и легко сканировать URL-адреса с помощью API. Вы можете сканировать не только веб-страницы, но и файлы, которые предоставляются вам для загрузки. Давайте рассмотрим некоторые инструменты API, которые помогут вам в этом. И, поскольку это API, усилия ваших разработчиков пойдут на пользу, если вы попросите их создать инструмент для сканирования веб-сайтов с использованием этих API.

Лучшие API для проверки на вирусы в 2024 году

Google Web Risk

Нет ничего удивительного в том, что программа для проверки веб-страниц исходит от компании, которая практически владеет

Интернетом (то есть всеми его веб-страницами). Google Web Risk довольно прост. Использовать API также очень просто. Чтобы проверить одну страницу с помощью командной строки, просто отправьте запрос следующим образом:

```
curl -H "Content-Type: application/json" "https://webrisk.googleapis.com/v1beta1/uris:search?key=YOUR_API_KEY&threatTypes=MALWARE&uri=http%3A%2F%2Ftestsafebrowsing.appspot.com%2Fs%2Fmalware.html"
```

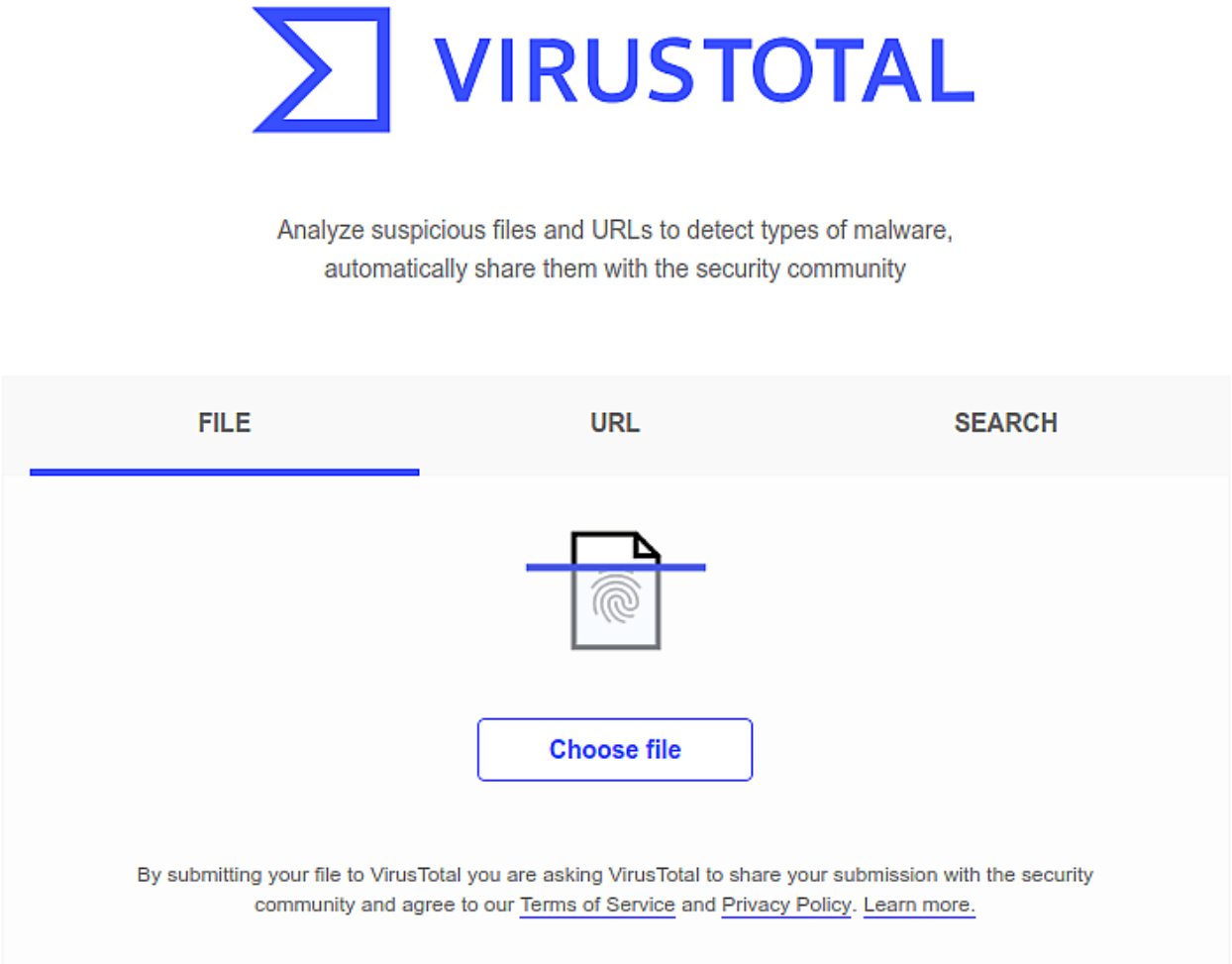
Если запрос прошел успешно, в ответ API сообщает тип уязвимости на странице:

```
{ "threat": { "threatTypes": ["MALWARE"], "expireTime": "2019-07-17T15:01:23.045123456Z" } }
```

Как видите, API подтверждает, что страница содержит вредоносное ПО. Обратите внимание, что Google Web Risk API не проводит диагностику URL или файла по требованию. Он обращается к черному списку, который ведет Google на основе результатов поиска и отчетов, и сообщает, находится ли URL в этом черном списке или нет. Другими словами, если этот API говорит, что URL безопасен, можно предположить, что он достаточно безопасен, но гарантий нет.

VirusTotal

VirusTotal – еще один классный сервис, который можно использовать для проверки не только URL-адресов, но и отдельных файлов (в этом смысле я ставлю его выше Google Web Risk по полезности). Если вам не терпится опробовать этот сервис, просто зайдите на его сайт, и прямо на главной странице вы найдете опцию для начала работы.



Хотя VirusTotal является бесплатной платформой, созданной и курируемой активным сообществом, он предлагает коммерческую версию своего API. Вот почему вам стоит заплатить за премиум-сервис:

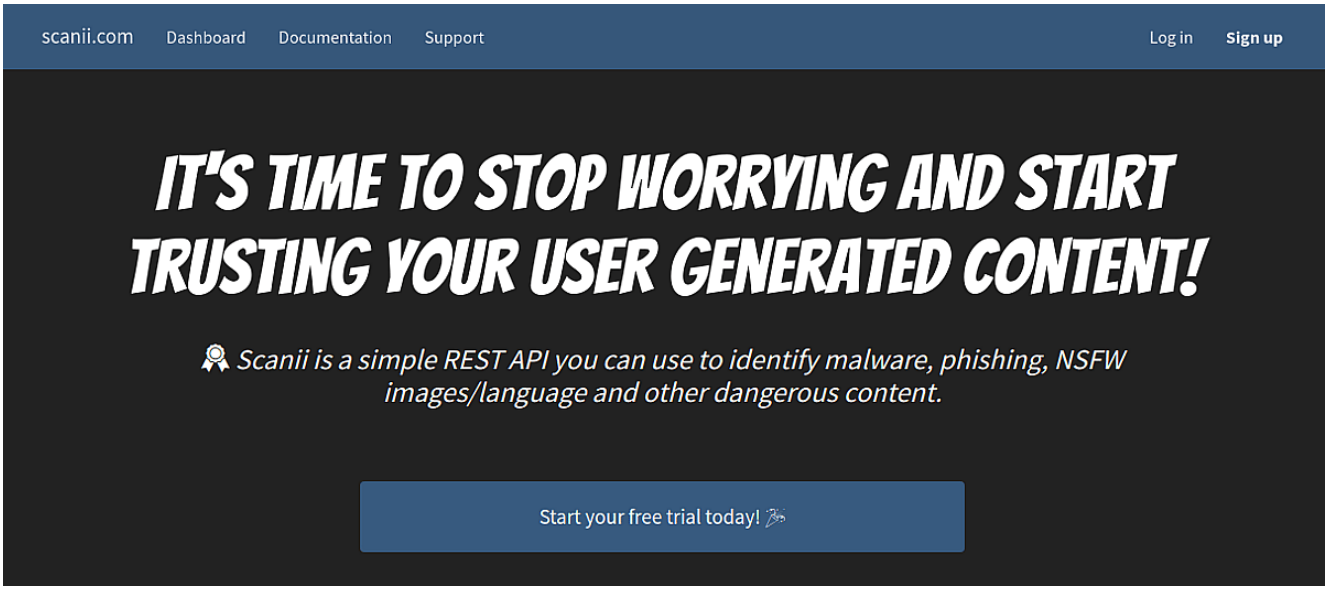
- Гибкая частота запросов и ежедневная квота (в отличие от всего лишь четырех запросов в минуту для публичного API)
- Отправленный ресурс проверяется антивирусом VirusTotal, после чего выдается дополнительная диагностическая информация.
- Поведенческая информация о файлах, которые вы отправляете (файлы будут помещены в различные “песочницы” для отслеживания подозрительных действий)
- Запрашивать базу данных файлов VirusTotal по различным параметрам (поддерживаются сложные запросы)
- Строгое SLA и время отклика (файлы, отправленные на

VirusTotal через публичный API, попадают в очередь и занимают значительное время для анализа)

Если вы воспользуетесь частным API VirusTotal, это может стать одной из лучших инвестиций в SaaS-продукт для вашего предприятия.

Scanii

Еще одна рекомендация по API-сканерам безопасности — Scanii. Это простой REST API, который может сканировать присланные документы/файлы на наличие угроз. Думайте о нем как о сканере вирусов по требованию, который можно запускать и масштабировать без особых усилий!



Вот что предлагает Scanii:

- Способность обнаруживать вредоносные программы, фишинговые скрипты, спам, контент NSFW (Not Safe For Work) и т. д.
- Он построен на базе Amazon S3 для легкого масштабирования и хранения файлов с нулевым риском.
- Обнаружение оскорбительных, небезопасных или потенциально опасных текстов на более чем 23 языках.
- Простой, без излишеств, целенаправленный подход к

сканированию файлов на основе API (другими словами, никаких излишне “полезных” функций)

Настоящим плюсом является то, что Scanii – это мета-движок; то есть он не выполняет сканирование самостоятельно, а использует набор базовых движков для выполнения работы. Это очень удобно, так как вам не нужно привязываться к конкретному движку безопасности, а значит, не нужно беспокоиться об изменениях в API и тому подобном. Я считаю Scanii огромным подспорьем для платформ, которые зависят от пользовательского контента. Еще один вариант использования – сканирование файлов, созданных поставщиком услуг, которому нельзя доверять на 100%.

Metadefender

Для некоторых организаций сканирования файлов и веб-страниц на одной конечной точке недостаточно. У них сложный информационный поток, и ни одна из конечных точек не может быть скомпрометирована. Для таких случаев Metadefender – идеальное решение. Воспринимайте Metadefender как параноидального привратника, который стоит между вашими основными информационными ресурсами и всем остальным, включая сеть. Я говорю “параноидальный”, потому что такова философия дизайна Metadefender. Я не могу описать ее лучше, чем они сами, так что вот:

> *Большинство решений по кибербезопасности полагаются на обнаружение как на основную защитную функцию. Санирование данных MetaDefender не полагается на обнаружение. Она предполагает, что все файлы могут быть заражены, и перестраивает их содержимое с помощью безопасного и эффективного процесса. Она поддерживает более 30 типов файлов и создает безопасные и пригодные для использования файлы. Санирование данных чрезвычайно эффективно для предотвращения целевых атак, программ-вымогателей и других типов известных и неизвестных вредоносных программ.*

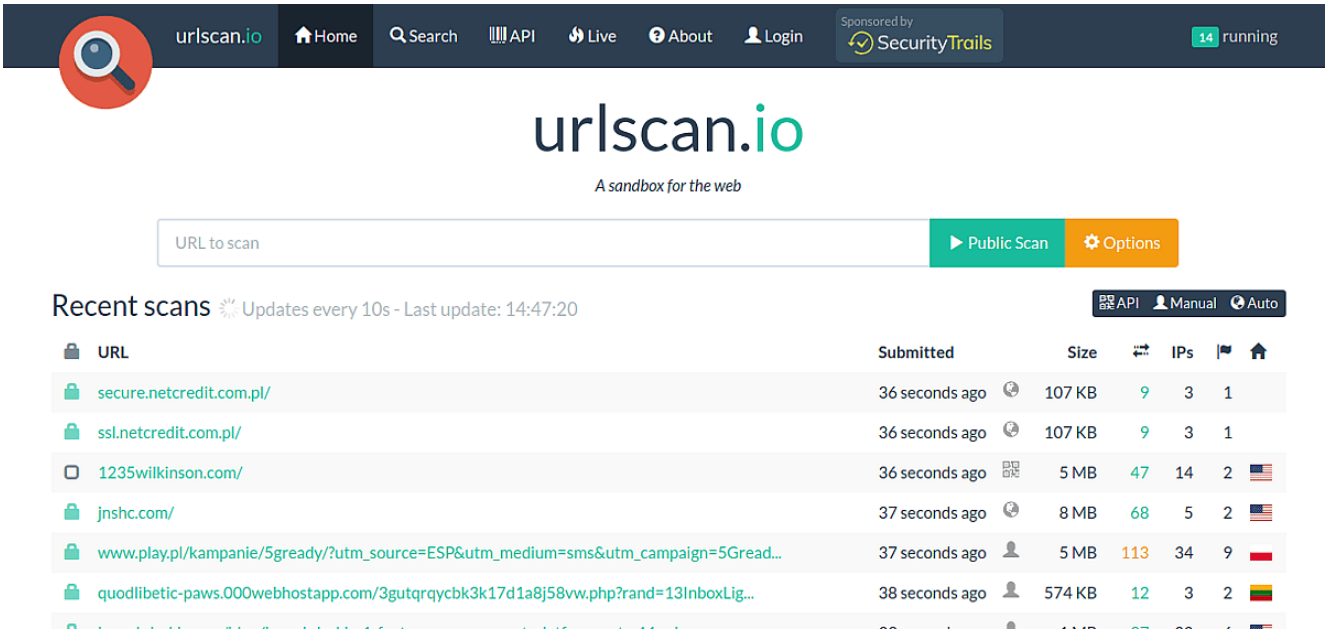
Metadefender предлагает несколько удобных функций:

- Предотвращение потери данных: Проще говоря, это способность отменять и защищать конфиденциальную информацию, обнаруженную в содержимом файлов. Например, квитанция в формате PDF с видимым номером кредитной карты будет обфусцирована Metadefender.
- Развертывание локально или в облаке (в зависимости от того, насколько вы параноик!).
- Просмотрите 30 с лишним типов форматов архивирования (zip, tar, rar и т.д.) и 4 500 трюков по подмене типов файлов.
- Многоканальное развертывание – защитите только файлы или перейдите к управлению электронной почтой, сетью и логином.
- Пользовательские рабочие процессы для применения различных типов конвейеров сканирования на основе пользовательских правил.

Metadefender включает в себя 30+ движков, но при этом хорошо абстрагируется от них, так что вам никогда не придется о них думать. Если вы относитесь к средним и крупным предприятиям, которые просто не могут позволить себе кошмары безопасности, Metadefender – отличный вариант.

Urlscan.io

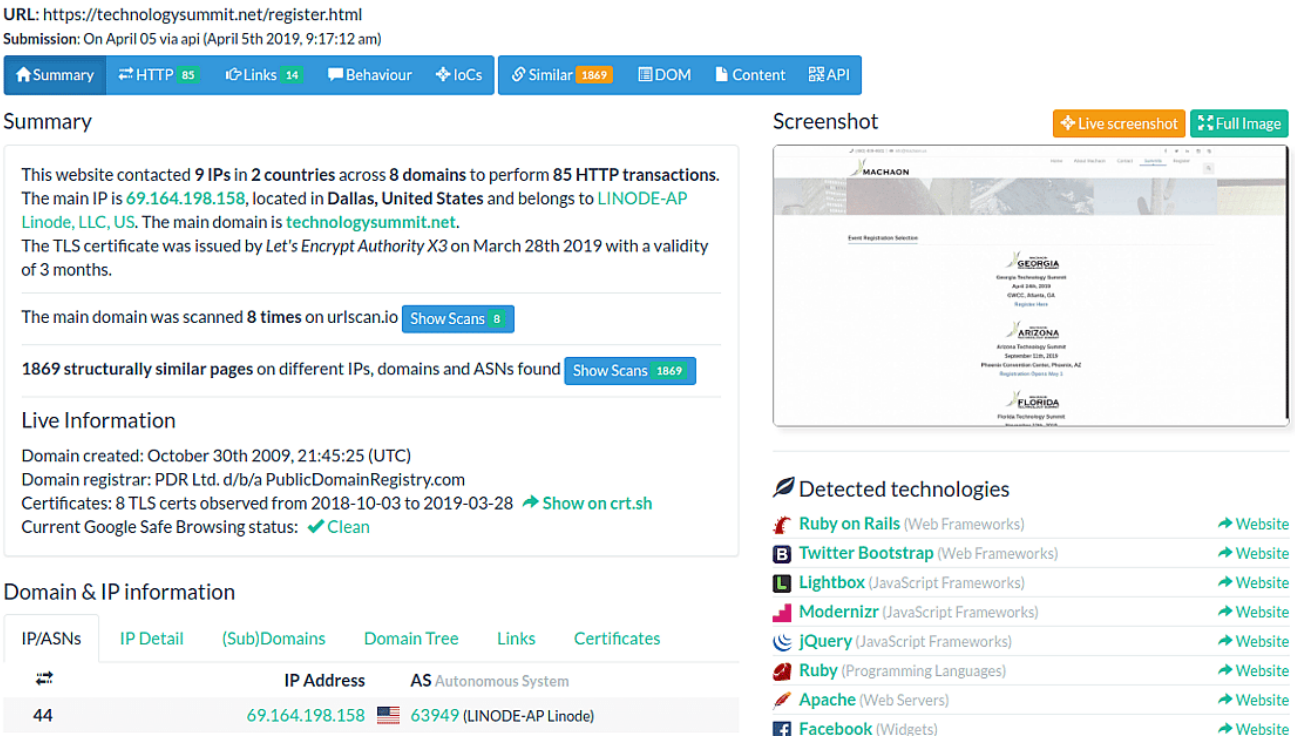
Если вы в основном работаете с веб-страницами и всегда хотели более детально изучить, что они делают за кулисами, Urlscan.io станет отличным оружием в вашем арсенале.



Объем информации, которую вываливает Urlscan.io, просто впечатляет. Среди прочего, вы сможете увидеть:

- Общее количество IP-адресов, к которым обращалась страница.
- Список географий и доменов, которым была отправлена информация со страницы.
- Технологии, используемые на фронтэнде и бэкэнде сайта (на точность не претендуем, но это тревожно точно!).
- Информация о домене и SSL-сертификате
- Подробные сведения о взаимодействии по протоколу HTTP с указанием полезной нагрузки запроса, имен серверов, времени ответа и многого другого.
- Скрытые перенаправления и неудачные запросы
- Исходящие ссылки
- Анализ JavaScript (глобальные переменные, используемые в скриптах, и т.д.)
- Анализ дерева DOM, содержание форм и многое другое.

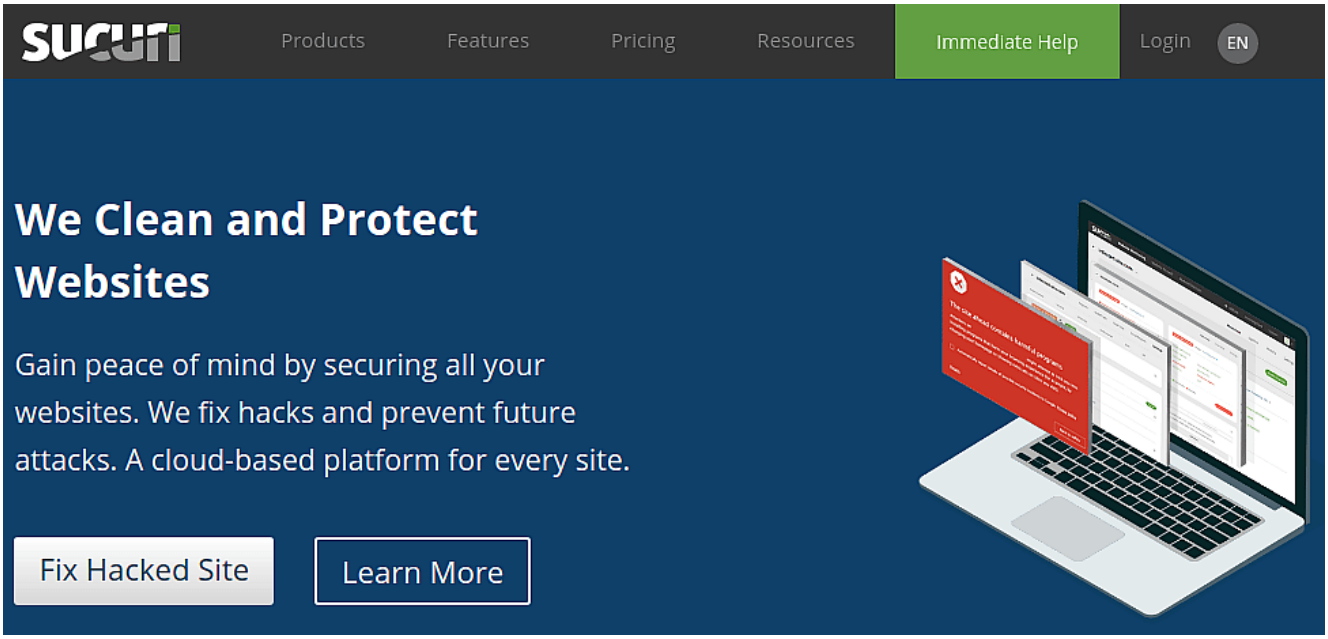
Вот как все это выглядит:



API прост и понятен, позволяет отправить URL для сканирования, а также проверить историю сканирования этого URL (то есть сканирования, выполненного другими людьми). В целом, Urlscan.io предоставляет массу информации для любого заинтересованного предприятия или частного лица.

SUCURI

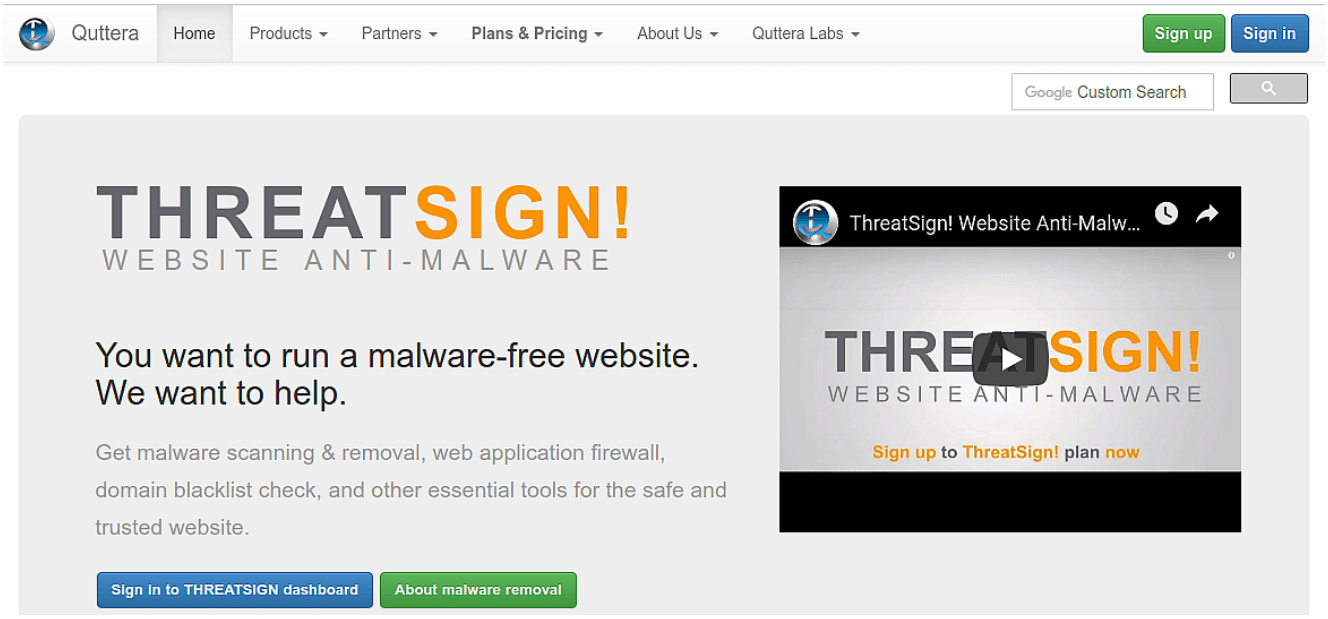
SUCURI – известная платформа для онлайн-сканирования веб-сайтов на предмет угроз и вредоносного ПО. Но вы, возможно, не знаете, что у них есть , позволяющий использовать те же возможности программно.



Говорить здесь особо не о чем, кроме того, что API прост и работает хорошо. Конечно, Sucuri не ограничивается API для сканирования, поэтому, пока вы здесь, я рекомендую вам ознакомиться с некоторыми из его мощных функций, таких как (по сути, вы предоставляете учетные данные FTP, а он входит в систему и сканирует все файлы на наличие угроз!)

Quttera

Последний в этом списке – Quttera, предлагающий несколько иной подход. Вместо того чтобы сканировать домен и представленные страницы по требованию, Quttera может осуществлять непрерывный мониторинг, помогая вам избежать уязвимостей нулевого дня.



REST API – простой и мощный, он может возвращать не только JSON, но и несколько других форматов (например, XML и YAML). При сканировании поддерживается полная многопоточность и параллельность, что позволяет запускать несколько исчерпывающих сканирований параллельно. Поскольку сервис работает в режиме реального времени, он неоценим для компаний, которые занимаются критически важными предложениями, где простой означает гибель.

Заключение

API-сканеры безопасности, подобные тем, о которых пойдет речь в этой статье, – это просто дополнительная линия защиты (или предосторожности, если хотите). Как и антивирусы, они могут многое, но они никак не могут обеспечить безотказный метод проверки. Это просто потому, что программа, написанная с вредоносным умыслом, для компьютера то же самое, что и программа, написанная для положительного воздействия – они обе запрашивают системные ресурсы и делают сетевые запросы; дьявол кроется в контексте, который не позволяет компьютерам успешно работать. Тем не менее, эти API обеспечивают надежную защиту, которая желательна в большинстве случаев – как для внешних сайтов, так и для ваших собственных!